



HZ Books

华章科技



# Blender

## 权威指南

### The Definitive Guide

罗聪翼 著

全球首本中文版Blender著作，Blender世界杯艺术挑战赛冠军亲自主笔，Blender之父作序推荐，权威性毋庸置疑！

全球20名顶尖Blender艺术家在本书中分享宝贵经验，BlenderCN中文社区官方推荐教程。

4大经典项目全工程详细制作解析，基础、建模、灯光、渲染、动画与特效全接触。

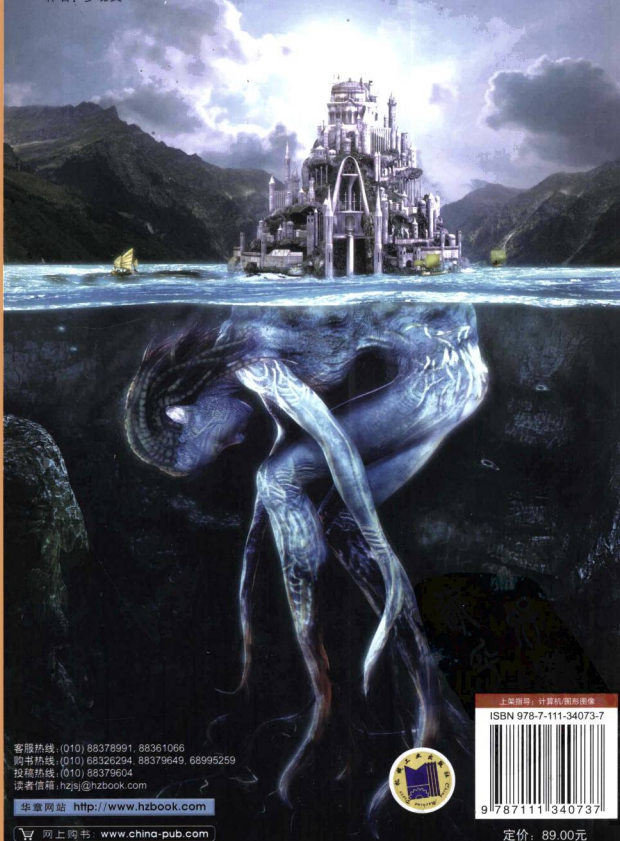
掌握基本专业技能，激活创作设计灵感，熟悉项目工作流程，积累实战经验技巧，实现数字艺术梦想。



机械工业出版社  
China Machine Press

作品名称: Floating (2009年Blender世界杯艺术挑战赛冠军作品)

作者: 罗晓翼



客服热线: (010) 88378991, 88361066  
购书热线: (010) 68326294, 88379649, 68995259  
投稿热线: (010) 88379604  
读者信箱: hzsj@hzbook.com

华章网站 <http://www.hzbook.com>

网上购书: [www.china-pub.com](http://www.china-pub.com)

上架指导: 计算机/图形图像

ISBN 978-7-111-34073-7



9 787111 340737

定价: 89.00元





# Blender

## 权威指南

### The Definitive Guide



Blender 是全球首款开源 3D 制作软件，也是最成功和最受欢迎的 3D 动画制作软件之一。本书是全球首本中文 Blender 著作，由国内资深 Blender 专家、Blender 世界杯艺术挑战赛冠军亲自执笔，Blender 之父自作序推荐，国内权威的 Blender 中文社区 BlenderCN 官方推荐教程。

本书内容系统而全面，详尽地讲述了 Blender 的各种功能和特性，并一步一步地讲解了它们的使用方法，可操作性极强；技术新颖，基于最新版的 Blender 撰写；实战性强，书中不仅设计了大量精巧的小案例，以及详细介绍了这些案例的实现过程，而且还包含 4 个工程性的大案例。本书不仅能满足你全面而系统地学习 Blender 理论知识的需求，还能满足你需要充分实践的需求。

全书分为三个部分：基础篇以全球 20 位顶尖 Blender 艺术家自述学习和使用 Blender 的经验开篇，全面地介绍了 Blender 的基本功能及其操作方法，包括 Blender 的安装与配置、基本的操作与界面、建模、修改器、灯光、材质与贴图、渲染等内容；高级篇讲解了 Blender 的一些高级特性和扩展用法，包括动画工具、物理特效与仿真、后期合成、Python 脚本扩展和游戏引擎等重要内容；实例篇以迭代的方式介绍了汽车建模与机械绑定、角色模型制作、室内效果图应用和角色动画等 4 个经典的工程性案例的制作方法过程，极具实践指导意义。

无论你是有经验的 Blender 用户，还是还没有经验的 Blender 爱好者，无论你是 CG 和 3D 动画爱好者/艺术家，还是 2D 平面艺术家，本书不仅能帮助你掌握基本的专业技能，熟悉项目工作的流程，还能激活你的创作设计灵感，积累实战经验技巧，最终实现你的数字艺术梦想。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

## 图书在版编目 (CIP) 数据

Blender 权威指南 / 罗聪翼著. —北京：机械工业出版社，2011.4

ISBN 978-7-111-34073-7

I. B… II. 罗… III. 三维—动画—图形软件，Blender IV. TP391.41

中国版本图书馆 CIP 数据核字 (2011) 第 057460 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑：陈佳媛

北京瑞德印刷有限公司印刷

2011 年 5 月第 1 版第 1 次印刷

185mm × 260mm · 41.5 印张

标准书号：ISBN 978-7-111-34073-7

定价：89.00 元


凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com



## Foreword 推荐序(一)

当罗聪翼先生邀请我为他的新书作一篇序时，按照我以往回应类似请求的习惯，我通常会先感谢这位作者的好意，然后再委婉地谢绝，就像在过去的5年内一样，我这样回复了每一位邀请我写序的朋友。因为现在有关 Blender 的图书已经相当多了，我甚至都数不清市面上已经有多少本相关的教程和书籍。如果要我为每一位作者都撰写一篇序的话，那工作量简直太大了！

但是我转回头来，又重新考虑了一下：虽然现在已经出版了数量这么多的 Blender 图书，但这是第一本正式的 Blender 的中文教程。这对我本人和每一位熟悉 Blender 的用户来说，都绝对是划时代的一步！

在欧洲和北美，Blender 已经得到了广泛的实际应用。我相信，对于这款自由开源的软件，亚洲绝对是一个充满更多机会和空间的市场，特别是在中国。开源技术可以为更多的独立艺术家、个人团体、学校或创业型企业带来潜在的巨大推动力，对于像中国这样的新兴经济体，开源和自由体系恰好可以让你获得更强的独立性和自主性。而且更重要的是，它可以允许你实现共享知识产权的建设和发展，让你能获得比同类封闭型产品进步更快速的能力。

Blender 是一款富有魅力的 3D 制作工具，同时也是一项充满了自由创意和开放源码的公共发展项目。它的进步完全取决于为它的开发做出贡献的软件工程师、使用它的艺术家，以及在线上或线下一起学习并互相激励，为成为优秀的艺术家或设计师而共同努力的 CG 爱好者们。与此同时，Blender 还是一款真正属于“为你量身定做”的软件，就像很多大公司都有一套内部开发的软件系统一样，你完全可以使用它来打造一个真正属于自己的 3D 创作平台。

我衷心地希望读者通过对本书的学习，能够打开心扉，实现思维的拓展，同时我也在这里热情地欢迎你加入这个全球性的艺术家和开发者团体。中国有着非常古老而美丽的文化，以及众多富有创造思维和时代风格的艺术家的。我真心地期待未来能看到更多由中国设计师使用 Blender 创作出来的美妙艺术品和精彩动画电影！

祝福你们可以在阅读本书的过程中寻找到更多的乐趣。

Ton Roosendaal

Blender 基金会主席

荷兰，阿姆斯特丹，2010 年 11 月



## Foreword 推荐序(二)

最初 Blender 只是一款用于开发 PS 游戏的内部工具，但是后来由于公司倒闭，它也差一点就被永远地封存起来。幸运的是，它很快就被由上万爱好者捐款成立的基金会所收购，并以这种特殊的方式，进行了一种生命形式的转变——由私有软件变成了源码公开、大家都可以为其添砖加瓦的开源软件。

初期的 Blender 在建模细分上相比同类产品更具有优势，它能够轻松实现多边形在高等级细分下的流畅编辑。紧接着在后来几次围绕短片而进行的集中开发过程中，Blender 的功能在逐渐被一一完善，特别是在角色动画方面有了长足的进步。与此同时，Blender 的多个功能也逐渐开始能满足行业上的一些应用。如最开始在角色动画方面使用的 UV 解算技术，就被广泛用在了游戏模型的制作上，很多游戏工作室都在其流水线中加入了 Blender，例如经典游戏《火炬之光》的建模开发。由于 Blender 的各种特效模块设计都十分方便而且易于使用，所以在一些小型的视觉演示中，也常使用 Blender 来分担一些特效的绘制工作。

作为一款开源软件，Blender 的功能变更是非常迅速和敏捷的，我们也可以理解为其代码和功能的新陈代谢速度非常快。开发者随时都可能因为有更好的功能模块而将原有的模块替换甚至删除掉。但是这样问题也出现了，所有以前编写出版的教程也将因为软件版本的更新而变得完全不可用。所以 Blender 虽然在功能和技术水平方面一直走在最前沿，但是在用户的知识积累上却产生了很多的困惑和麻烦。例如最新的 2.5 版已经发布一年多了，但是当用户在网上寻找相关功能介绍时，得到的搜索结果却是屈指可数，甚至连官方的 Wiki 手册都没有写进这一次的全部更新。而且相比于早期版本（例如 2.4）的内容，两个版本之间包括快捷键或者按钮图标都已经完全不一样了！

由于 Blender 的特殊开发背景，它在计算机图形学和物理模拟学等方面的技术，很快就以科研的方式引进到了国内，并被国内部分大学实验室用于相关领域的研究。而在实际应用中，Blender 和 SIO2 引擎之间结合得非常好，使用 Blender 制作的工程可以直接在 SIO2 中执行读取等操作，所以还在 iPhone 和 iPad 的 3D 游戏方兴未艾之时，国内就已经有很多个人开始将 Blender 用于开发 iPhone 手机游戏应用中。现在，国内的一些工作室也已经将 Blender 作为主要的制作工具，整合进了高清数字电影的创作流程当中，有的是在工作流的一个环节中使用 Blender，也有的是在工作流的多个环节中使用它。据不完全统计，Blender 在中国大陆的使用情况还处于初中级阶段，并且大部分都集中在京、沪、广三地，其中研发等科研性应用集中在北京，而上海、广州和深圳都逐渐引入了一定程度的商用模式。

早在 2004 年的时候，我们就正式成立了 BlenderCN 中文社区，目的是在国内推广 Blender，并提供及时的本地化支持。记得成立初期 Blender 还是刚开源的 2.1 版本，基金会也刚经历了开发人员从全部流失到重启开发的一个过程。BlenderCN 社区中的朋友都是自发地聚在一起，利用业余时间一边翻译 Blender 的 Wiki 文档，一边在讨论中共同学习 Blender。不到半年的时间，我们就完成了 2.34 版本的 Wiki 翻译，同时也完成了 2.4 系列版本的汉化工作。可以说，BlenderCN 是国内唯一一直坚持到现在的 Blender 咨询、培训和技术交流社区团队。

有使用 and 市场需求就需要有培训和知识转移，但是由于缺乏一本标准的教材，相关的从业人员同国外的使用者比起来，还有相当大的一段差距，同时非正式培训人员的能力评估标准化也亟待解决。就 Blender 本身的功能来说，差不多涵盖了当今通用数字内容创建工具的所有门类，对于如此庞大的体系，如果没有一本好的参考书在手边，一定会手忙脚乱。而现如今，Blender 的成体系书籍很多，不过能找到的大都是英文版，这为国内的学习爱好者增加了不少难度。虽然在社区中，我们会有根据项目经验编制中文视频教程等资源的计划，但是从很多用户和公司的反馈来看，整个中文 Blender 用户群对一本中文教程书籍的需求迫切性还是相当高的，希望罗聪翼先生的这本书能为改善这种状况带来巨大的推动作用。

这本书从策划到编写和校对，花了一年多的时间。一年时间可以做很多事情，不过罗聪翼先生选择了坚守，把他的所知所学都寄存到笔下，为后来者留下一块敲门砖。我和罗聪翼先生是在 2007 年底认识的，他也是在那时才刚开始学习 Blender，我和社区中的所有朋友都一同见证着他的进步。他敢于不断地去挑战自己，不断地尝试创作作品，并且积极地参加社区比赛，丰富自己的实战经验，实现自己对 CG 这个概念的认识。努力追求完美的他，获得的奖项和认同也颇丰，所以国内的第一本中文 Blender 教程由他来执笔，理所应当。

本书出版后，BlenderCN 的朋友们也将和罗聪翼先生一起，在社区为本书提供相应的技术支持和问题解答。欢迎你加入 BlenderCN 的大家庭，让我们在这种别样的开源社区中，不断地完善自我，不断地学习成长吧！

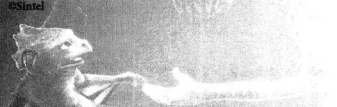
祝各位都能快乐地享受 Blender！

裴雪柯 (Kidux)

BlenderCN 中文社区创始人

中国，成都，2010 年 12 月





## Preface 前言

### 为什么要写这本书

几年前当我开始学习 CG 的时候，不少朋友都向我建议，去试试那些好莱坞都在用的工具吧！于是我开始去图书馆翻书找资料，甚至去咨询了一些专业培训机构。但是我很快便意识到自己犯了初学者常犯的错误：迷信工具的力量！有经验的艺术家都知道，作品的重点不是技术，而在于人。好的工具可以辅助你完成优秀的作品，但是它最终只是一种表面现象，人与软件之间的沟通过程才是创作的本质。

大学图书馆为我带来了许多美好的回忆，其中之一就是它提供了大量的免费杂志。在一次偶然翻阅美术杂志的时候，我看到了一篇介绍开源电影《Elephants Dream》的文章。一方面我本身就是开源产业的忠实拥护者，另一方面我也厌倦了对大型软件近乎绝望的学习，当然也有我的电脑太过陈旧的因素，那台又破又老的主机根本无法运行庞大的主流制作软件。于是我开始尝试这款只有 2MB，甚至连任何中文资料都没有的软件。刚开始，我认为 Blender 只不过是一种傻瓜版的工具，至少可以让我做点像样的东西出来，而且从当时的角度来看，它也的确如此。但是随着时间流逝，以及对它的深入了解，我才明白，这个软件是我见过的最特别的一款软件，与其他工具完全不同！

Blender 体现了设计者对它的终极设计目标：为用户降低复杂度，为工作提高效率。开源设计模式是对 Blender 进步影响最深远的因素之一，因为为用户设计软件总是非常困难，开发者通常无法察觉艺术家对工具操作的要求和希望。而来自开源社区中及时的用户反馈，使得整个过程变得更简单，设计者们可以通过更有效率、更廉价的方式与使用者直接进行沟通，革命性的开源模式让这一切事情变得如此神奇而简单。

我很欣赏这种模式，学习非共享而不能进步。当你无法负担高昂的培训费，而在一个人的盲目学习中感到失落与无助时，或者当你曾经希望多接触真正的艺术家和专业的工业资源，却无路可走的时候，我认为你可以来试着加入开源社区，学习一下 Blender，体验这种人与人之间互动的新兴学习模式。多读书，多交流，志同道合的朋友在一起更容易彼此打气加油，还可以互相切磋，学习很快就能变为一种享受。

Blender 还在不断地演化，它引导的是一场真正的学习革命，是一种全球化思维的碰撞，是一种突破传统封闭式交流的海量数据共享。也许 Blender 还不是这场革命的导火索，但是至少我认为，能够带领大家去学习和认识这种思维，也是一件非常有意义的事情。

作为 BlenderCN 社区监制的第一本中文教程，社区网站 <http://blendercn.org> 和 <http://blog.blendercn.org> 将为读者提供相应的 Blender 知识管理和及时的答疑解惑支持，以及国内外最新的 Blender 资讯。我们的社区是一个非盈利性的民间组织，所有成员都是为了推动 Blender 在国内的发展，而自愿团结在一起，义务地为公众奉献服务。我很感谢社区这么多年来对我的帮助，也敬佩大家为开源事业作出贡献的精神。写书的这一年全靠兴趣为动力，我也并不想以此牟利，所以决定将本书带来的所有收入全部捐献给 BlenderCN 开源社区，用于支持和维护社区今后的发展。

## 本书目标

每一个最新版本的 Blender 都会被下载超过百万次，这个数字还不包括下载源码做自行编译的用户。Blender 在全球范围内的使用者也接近上千万，并且每天都有更多的用户开始学习或者转型使用 Blender。

在国外，特别是针对个人艺术家或小型工作室，Blender 正在发挥着病毒式的传播影响力。在国内，一部分先行的公司已经开始逐渐引进 Blender 的工业生产线，将 Blender 的思维作为一种全新的理念带入国门。

本书希望能提供一个从入门到精通的学习过程，无论是初学 3D 的爱好者，还是行业从业人员，我希望你们都能从本书获益。

## 读者对象

这里我们根据软件需求划分出一些能使用 Blender 的用户团体，这些用户都是本书潜在的读者群。

- ☐ Blender 用户和爱好者
- ☐ Blender 代码开发自愿者
- ☐ CG 和 3D 动画爱好者
- ☐ 2D 平面艺术家或者相关平面设计工作室
- ☐ 3D 动画艺术家或者相关动画广告工作室
- ☐ 使用 Blender 参与工作流程的公司与集体
- ☐ 开设相关课程的大专院校

## 如何阅读本书

本书分为三大部分，其中第三部分以接近实战的实例来讲解工程应用，相比于前两部分更独立。如果你是一名经验丰富的资深用户，能够理解 Blender 的相关基础知识和使用技巧，那么你可以直接阅读这部分内容。但如果你是一名初学者，请一定从第 1 章的基础理论知识开始学习。

第一部分为基础篇，简单介绍 Blender 的基本使用技巧和相关理论，帮助读者了解一些基础

背景知识，并熟悉 Blender 的全新操作模式。

第二部分为高级篇，着重讲解 Blender 的部分高级应用功能，包括动画工具、物理模拟和后期结点系统等模块。

第三部分为实例篇，通过对 4 个不同风格的典型案例进行讲解，让读者能够了解一个完整的 Blender 生产流程。

附录 A 为 Blender 和 BlenderCN 的年谱大事记。

附录 B 为 Blender 2.5 以及最新版本的默认快捷键列表。

## Blender 版本

Blender 的发布更新以 2.5X 的小版本号来表示，本书针对的是 2.56 版本。相比早期的版本，这个版本的改动很大，弥补了大量的缺陷，引入了很多优秀的新功能，与 2.4 系列存在巨大差异。对于一些成熟的生产模式，例如 iPhone 游戏开发，你可能还是需要转回使用 2.49 版本，因为一些生产配套的第三方软件接口可能还没有完成对新版 2.5API 的更新。

## API 文档

Blender 基金会维护着 API 的更新，读者可以登录开发页面<sup>①</sup>去查看对应版本的在线文档。其中 2.49 版本的 API 类库最完整，而 2.5 版本的 API 还处于更新完善阶段，部分功能还在开发和重写中，所以本书不再重复这份文档。

## 源代码

本书的所有源代码和工程文件都以 Creative Commons Attribute 3.0 创作共享许可协议的形式发布，你可以自由使用和学习分享，如果需要应用于商业领域，请注明版权所有。如果你行使本许可授予的使用源代码的权利，就表明你接受并同意遵守本许可的条款，对其使用不得超越本许可授权的范围。

## 勘误和支持

由于作者的水平有限，编写的时间也很仓促，书中难免会出现一些错误或者不准确的地方，不妥之处在所难免，恳请读者批评指正。为此，我特意创建了一个在线支持与应急方案的二级站点 <http://book.blendercn.org>。你可以将书中的错误发布在 Bug 勘误表页面中。同时如果你遇到任何问题，也可以访问 Q&A 页面，我将尽量在线上为读者提供最满意的解答。书中的全部源文件都将发布在这个网站上，我也会将相应的功能更新及时发布出来。如果你有更多的宝贵意见，也

① 网站地址：<http://www.blender.org/development/>。

欢迎你发送邮件至我的邮箱 [congcong009@Gmail.com](mailto:congcong009@Gmail.com)，我很期待能够听到你们的真挚反馈。

## 关于电子版

本书中有很多精美的效果图，但是很遗憾，由于考虑到成本和定价等因素，本书未能印刷成彩色。为了便于大家既能以很便宜的价格买到这本书，又能欣赏到书中的美丽图片，我们将同步推出该书的电子版，如果大家有需要，欢迎到中国互动出版网（[www.china-pub.com](http://www.china-pub.com)）购买。

## 致谢

我首先要感谢伟大的 3D 动画精神领袖 Ton，他开创了一款影响我整个人生的软件。

感谢南京航空航天大学图书馆，它为我大学四年提供了一个轻松而充实的学习环境，还有数不清看不完的书籍和杂志。

感谢 BlenderCN 社区中每一位充满创意和活力的朋友——Kidux、DeathBlood、冰冻牡蛎、Harrison、莫克陶、水经石、sumomofan、Mike Pan、中原狼、小无花果他爹、Fox、Steven (ChinaVFX)、疯子小林、月光光、老猫、Johnny 孙、zxfyzz、wiilovy、Jollin、遥远的桥、wewe 猫、coffee、Magic 骏、金鱼、蜗牛的邻居、icebird、blackorange、绵羊小菲、菜肉包、ewalker、城主、铁毛、青春森林，以及这个仓促写就的名单之外的更多朋友，感谢你们长期对社区的支持和贡献。感谢李锟老师的引荐，在你的努力下才促成了这本书的出版。

感谢机械工业出版社华章公司的编辑杨福川老师，感谢你的魄力和远见。在这一年多的时间中你始终支持我写作，是你的鼓励和帮助引导我顺利完成全部书稿。

最后我一定要感谢我的爸爸、妈妈、姥姥、姥爷、小姨、婆婆、爷爷，感谢你们将我培养成人，并时时刻刻为我灌输着信心和力量！

谨以此书，献给我最亲爱的家人，以及众多热爱 Blender 的朋友们。

罗聪翼 (congcong009)

中国，南京，2010 年 12 月



# Conternt 目 录

推荐序 (一)

推荐序 (二)

前 言

## 第一部分 基础篇

### 第1章 Blender 概述 ..... 3

#### 1.1 开源简介 ..... 3

1.1.1 Blender 是什么 ..... 3

1.1.2 Blender 的发展与历史 ..... 3

1.1.3 社区支持 ..... 6

#### 1.2 使用 Blender 的艺术家们 ..... 8

1.2.1 Andrew Price ..... 8

1.2.2 Anna Celarek ..... 10

1.2.3 Barath Endre ..... 12

1.2.4 Ben Dansie ..... 12

1.2.5 Colin Levy ..... 14

1.2.6 Enrico Valenza ..... 15

1.2.7 Kamil Kuklo ..... 16

1.2.8 Kevin Hays ..... 17

1.2.9 Mathias Helmuth Pedersen ..... 18

1.2.10 Matt Ebb ..... 20

1.2.11 Mike Pan ..... 23

1.2.12 Pablo Vazquez ..... 24

1.2.13 Robert J. Tiess ..... 25

1.2.14 Sebastian König ..... 26

1.2.15 Soenke Maeter ..... 27

1.2.16 Tony Mullen ..... 28

1.2.17 Vitor Balbio ..... 30

1.2.18 Диметр Калинин (Dmitry  
Kalinin) ..... 30

1.2.19 黄海 ..... 31

1.2.20 王永海 ..... 33

#### 1.3 Blender 的安装与配置 ..... 35

1.3.1 系统要求 ..... 35

1.3.2 快速安装 ..... 37

1.3.3 本地编译 ..... 37

### 第2章 基本操作与界面 ..... 40

#### 2.1 Blender 基础概念与基本操作 ..... 40

2.1.1 鼠标操作与键盘常用  
功能键 ..... 40

2.1.2 物体与原心 ..... 40

2.1.3 Blender 的界面操作 ..... 41

#### 2.2 用户参数设置 ..... 43

2.2.1 用户界面设置 ..... 43

2.2.2 编辑设置 ..... 44

2.2.3 输入控制设置 ..... 46

2.2.4 附加脚本设置 ..... 48

2.2.5 主题设置 ..... 48

2.2.6 文件系统 ..... 49

2.2.7 系统设置 ..... 50

#### 2.3 窗口类型 ..... 52

2.3.1 三维视图 ..... 52

2.3.2 时间轴窗口 ..... 63

2.3.3 曲线图形编辑器 ..... 65

2.3.4 动作编辑器 ..... 69



2.3.5 非线性动画编辑器 .....	70	3.5 曲线建模 .....	121
2.3.6 图片编辑器 .....	73	3.5.1 曲线的编辑 .....	121
2.3.7 视频序列编辑器 .....	75	3.5.2 Bezier 曲线 .....	124
2.3.8 文字编辑器 .....	77	3.5.3 NURBS 曲线 .....	125
2.3.9 结点编辑器 .....	79	3.6 文字建模 .....	126
2.3.10 逻辑编辑器 .....	81	3.6.1 文字的输入 .....	126
2.3.11 工具面板 .....	81	3.6.2 文字的编辑与修改 .....	127
2.3.12 大纲管理器 .....	81		
2.3.13 信息窗口 .....	82	<b>第4章 修改器 .....</b>	<b>129</b>
2.3.14 文件浏览器 .....	84	4.1 结构类修改器 .....	130
2.3.15 控制台窗口 .....	85	4.1.1 阵列修改器 .....	130
<b>第3章 建模 .....</b>	<b>86</b>	4.1.2 倒角修改器 .....	135
3.1 新建物体 .....	87	4.1.3 布尔运算修改器 .....	137
3.1.1 网格物体 .....	87	4.1.4 构造修改器 .....	138
3.1.2 曲线物体 .....	88	4.1.5 精简修改器 .....	139
3.1.3 骨骼/空物体/摄像机物体 .....	89	4.1.6 硬边修改器 .....	139
3.1.4 灯光物体 .....	89	4.1.7 遮罩修改器 .....	140
3.2 编辑模式 .....	90	4.1.8 镜像修改器 .....	142
3.2.1 点线面结构 .....	90	4.1.9 细分修改器 .....	144
3.2.2 法线结构 .....	92	4.1.10 实体化修改器 .....	144
3.2.3 选择菜单 .....	92	4.1.11 多重解析修改器 .....	146
3.2.4 网格菜单 .....	95	4.1.12 UV 投影修改器 .....	147
3.3 基础网格建模 .....	106	4.1.13 螺旋修改器 .....	147
3.3.1 网格快捷工具栏 .....	106	4.2 形变类修改器 .....	148
3.3.2 平滑工具 .....	106	4.2.1 骨骼修改器 .....	148
3.3.3 使用关联复制实现镜像 建模 .....	107	4.2.2 塑性修改器 .....	148
3.3.4 螺旋和细分工具 .....	108	4.2.3 曲线修改器 .....	149
3.3.5 精确切割工具 .....	109	4.2.4 置换修改器 .....	150
3.3.6 基本建模工具的配合使用 范例 .....	109	4.2.5 晶格修改器 .....	152
3.4 高级网格建模 .....	112	4.2.6 钩链修改器 .....	153
3.4.1 顶点组 .....	112	4.2.7 网格造型修改器 .....	154
3.4.2 权重绘制 .....	113	4.2.8 收缩修改器 .....	155
3.4.3 雕刻模式 .....	114	4.2.9 快速形变修改器 .....	157
		4.2.10 平滑修改器 .....	158
		4.2.11 波浪修改器 .....	159
		4.3 模拟类修改器 .....	160

4.3.1	布料修改器 .....	160	6.1.2	材质选项 .....	197
4.3.2	碰撞修改器 .....	160	6.1.3	多材质应用 .....	199
4.3.3	爆破修改器 .....	160	6.2	着色器选项 .....	200
4.3.4	流体修改器 .....	161	6.2.1	漫反射着色器 .....	200
4.3.5	粒子实体修改器 .....	161	6.2.2	高光着色器 .....	203
4.3.6	粒子系统修改器 .....	163	6.2.3	渐变着色选项 .....	205
4.3.7	烟雾/软体模拟修改器 .....	163	6.2.4	光线追踪反射 .....	206
4.3.8	光线追踪折射 .....	163	6.2.5	光线追踪透明 .....	208
4.3.9	粒子系统修改器 .....	163	6.2.6	子面散射着色器 .....	212
4.3.10	粒子系统修改器 .....	163	6.2.7	体积着色器 .....	214
4.3.11	粒子系统修改器 .....	163	6.2.8	光晕着色器 .....	216
4.3.12	粒子系统修改器 .....	163	6.2.9	顶点着色器 .....	217
5	灯光 .....	164	6.3	纹理贴图 .....	218
5.1	光线的属性 .....	164	6.3.1	通用贴图参数 .....	219
5.1.1	光源属性 .....	165	6.3.2	映射输入 .....	222
5.1.2	光线衰减 .....	166	6.3.3	通道输出 .....	226
5.1.3	贴图灯光 .....	171	6.4	UV 贴图 .....	229
5.1.4	体积光 .....	171	6.4.1	UV 坐标系 .....	229
5.1.5	灯光的常规控制面板 .....	173	6.4.2	UV 解算和图层管理 .....	230
5.2	阴影类型 .....	173	6.4.3	UV 编辑器 .....	233
5.2.1	光线追踪阴影 .....	174	6.4.4	贴图绘制模式 .....	237
5.2.2	缓冲阴影 .....	175	6.4.5	网格 UV 拆分实例 .....	239
5.3	灯光类型 .....	180	7	渲染 .....	241
5.3.1	点灯光 .....	180	7.1	渲染设置 .....	241
5.3.2	日光 .....	181	7.1.1	着色选项 .....	242
5.3.3	聚光灯 .....	185	7.1.2	规格选项 .....	243
5.3.4	半球灯 .....	186	7.1.3	运动模糊选项 .....	245
5.3.5	面光源 .....	187	7.1.4	抗锯齿 .....	245
5.4	环境属性 .....	188	7.1.5	烘焙 .....	246
5.4.1	环境与背景 .....	189	7.1.6	命令行渲染 .....	248
5.4.2	环境光吸收 .....	189	7.2	输出设置 .....	250
5.4.3	环境光照 .....	192	7.2.1	格式选项 .....	250
5.4.4	间接光照 .....	192	7.2.2	性能选项 .....	252
5.4.5	迷雾与星辰 .....	193	7.2.3	后期合成选项 .....	253
5.4.6	HDRI 照明 .....	194	7.2.4	标记选项 .....	254
5.5	色彩管理和线性工作流 .....	194			
6	材质与贴图 .....	196			
6.1	着色器 .....	196			
6.1.1	材质的渲染原理 .....	196			

7.3 层输出属性 .....	254
7.3.1 分层渲染 .....	254
7.3.2 通道输出 .....	258
7.4 摄像机 .....	262
7.4.1 显示面板 .....	263
7.4.2 镜头面板 .....	263
7.5 网络渲染 .....	264
7.5.1 渲染农场 .....	264
7.5.2 服务器端 .....	266
7.5.3 负载端 .....	267
7.5.4 工作站端 .....	267
7.5.5 搭建实例 .....	267
7.6 如何提升渲染的性能 .....	270
7.6.1 工作环境 .....	270
7.6.2 工程配置 .....	270

## 第二部分 高级篇

### 第8章 动画工具 .....

8.1 基本动画工具 .....	275
8.1.1 曲线编辑器 .....	275
8.1.2 创建关键帧 .....	276
8.1.3 编辑曲线 .....	277
8.1.4 通道管理 .....	279
8.1.5 时间轴 .....	279
8.2 物体动画 .....	280
8.2.1 关键帧动画 .....	280
8.2.2 路径动画 .....	281
8.3 形变动画 .....	283
8.3.1 形变关键帧 .....	283
8.3.2 形变驱动器 .....	286
8.3.3 晶格形变 .....	287
8.3.4 钩铰形变 .....	289
8.4 骨骼系统 .....	289
8.4.1 编辑模式 .....	290
8.4.2 姿势模式 .....	292

8.4.3 骨骼物体数据选项 .....	293
8.4.4 骨头选项 .....	296
8.5 骨骼动画 .....	299
8.5.1 动作编辑器 .....	299
8.5.2 非线性编辑器 .....	299
8.6 约束控制 .....	300
8.6.1 变形约束 .....	300
8.6.2 跟随约束 .....	305
8.6.3 关系约束 .....	311

### 第9章 物理特效与仿真 .....

9.1 粒子仿真系统 .....	315
9.1.1 粒子系统 .....	316
9.1.2 发射器选项 .....	317
9.1.3 物理效果选项 .....	319
9.1.4 可视化选项 .....	325
9.1.5 渲染选项 .....	325
9.1.6 速率控制选项 .....	330
9.1.7 旋转控制选项 .....	331
9.1.8 子粒子系统选项 .....	331
9.1.9 毛发选项 .....	334
9.2 烟雾仿真系统 .....	338
9.2.1 烟雾域选项 .....	339
9.2.2 烟雾发射体选项 .....	339
9.2.3 碰撞与障碍选项 .....	340
9.2.4 烟雾渲染选项 .....	341
9.3 软体仿真系统 .....	342
9.3.1 软体工作原理 .....	342
9.3.2 软体的外部作用力 .....	343
9.3.3 软体的碰撞计算 .....	345
9.3.4 软体的内部作用力 .....	347
9.4 布料仿真系统 .....	351
9.4.1 布料初始化 .....	351
9.4.2 布料碰撞选项 .....	352
9.4.3 布料硬度控制选项 .....	353
9.4.4 外力场选项 .....	353
9.4.5 烘焙选项 .....	355

9.5 流体仿真系统 .....	355	10.4.7 影像类结点 .....	411
9.5.1 流体原理 .....	356	10.4.8 变形类结点 .....	414
9.5.2 流体域设置 .....	356	10.5 视频序列编辑器 .....	415
9.5.3 流体设置 .....	359	10.5.1 序列编辑器界面 .....	415
9.5.4 交互控制 .....	361	10.5.2 序列编辑 .....	418
9.5.5 流体仿真优化 .....	363	10.5.3 特效 .....	419
9.6 刚体仿真系统 .....	364	10.5.4 序列输出 .....	421
9.6.1 Bullet 引擎简介 .....	364		
9.6.2 环境配置 .....	365	<b>第 11 章 Python 脚本扩展</b> .....	422
9.6.3 物理选项 .....	365	11.1 Python 脚本基础 .....	422
9.6.4 仿真与记录 .....	366	11.1.1 Python 简介 .....	422
		11.1.2 编辑与编译 .....	423
<b>第 10 章 后期合成</b> .....	368	11.2 Blender 的 Python API .....	424
10.1 结点系统 .....	368	11.2.1 API 基础 .....	424
10.1.1 结点编辑器 .....	368	11.2.2 事件查看 .....	425
10.1.2 结点基础 .....	370	11.2.3 数据访问 .....	425
10.2 着色器结点模式 .....	372	11.2.4 操作符 .....	426
10.2.1 输入类结点 .....	372	11.2.5 正文 .....	429
10.2.2 输出类结点 .....	376	11.2.6 类型 .....	432
10.2.3 色彩类结点 .....	376	11.2.7 集成 .....	433
10.2.4 矢量类结点 .....	386	11.3 内置脚本 .....	437
10.2.5 转换类结点 .....	388	11.3.1 插件脚本 .....	437
10.3 纹理类结点 .....	393	11.3.2 安装插件脚本 .....	438
10.3.1 输入类结点 .....	393	11.3.3 齿轮生成脚本 .....	438
10.3.2 输出类结点 .....	395	11.3.4 云生成脚本 .....	439
10.3.3 色彩类结点 .....	395	11.4 外置程序脚本 .....	441
10.3.4 图案类结点 .....	395	11.4.1 YafaRay .....	441
10.3.5 内置纹理结点 .....	396	11.4.2 LuxRender .....	442
10.3.6 转换类结点 .....	397	11.4.3 Indigo .....	443
10.3.7 扭曲类结点 .....	397	11.4.4 VRay .....	443
10.4 合成类结点 .....	398	11.4.5 SunFlow .....	444
10.4.1 输入类结点 .....	398	11.4.6 Octane .....	445
10.4.2 输出类结点 .....	399		
10.4.3 色彩类结点 .....	400	<b>第 12 章 游戏引擎</b> .....	446
10.4.4 矢量类结点 .....	404	12.1 BGE 开发环境 .....	446
10.4.5 滤镜类结点 .....	405	12.1.1 BGE 系统 .....	447
10.4.6 转换类结点 .....	409	12.1.2 GLSL 高阶着色器 .....	448

12.1.3	逻辑编辑器 .....	450	13.2	车头建模 .....	480
12.1.4	Hello Game! .....	451	13.2.1	使用镜像修改器代替对称 建模 .....	480
12.2	物理属性 .....	455	13.2.2	制作车头部分 .....	481
12.2.1	无碰撞类 .....	456	13.3	车身建模 .....	489
12.2.2	静态类 .....	456	13.3.1	车身布线思路 .....	489
12.2.3	动态类 .....	456	13.3.2	车身建模 .....	490
12.2.4	刚体类 .....	458	13.3.3	车窗建模 .....	496
12.2.5	软体类 .....	458	13.4	附件建模 .....	497
12.2.6	闭塞类 .....	459	13.4.1	挤压厚度 .....	497
12.2.7	触发类 .....	461	13.4.2	制作车头护栏 .....	498
12.3	触发器 .....	463	13.4.3	制作车灯 .....	500
12.3.1	触发器结构 .....	463	13.4.4	制作车头网格护栏 .....	502
12.3.2	常用触发器 .....	465	13.4.5	制作车门把手 .....	504
12.4	控制器 .....	466	13.4.6	制作后视镜 .....	505
12.4.1	控制器逻辑 .....	466	13.5	车轮建模 .....	506
12.4.2	Python 逻辑控制器 .....	467	13.5.1	制作轮毂 .....	506
12.5	促动器 .....	469	13.5.2	利用修改器制作轮毂 镜像 .....	509
12.5.1	运动促动器 .....	469	13.5.3	制作轮胎 .....	511
12.5.2	曲线动画促动器 .....	470	13.6	硬边处理 .....	513
12.5.3	场景促动器 .....	470	13.6.1	使用切割细分来制作 硬边 .....	513
12.5.4	编辑促动器 .....	471	13.6.2	折痕工具制作硬边 .....	515
12.5.5	镜头促动器 .....	471	13.7	车轮机械约束设计 .....	517
12.5.6	声效促动器 .....	471	13.8	材质设计 .....	518
12.6	游戏开发与性能优化 .....	472	13.9	灯光设计 .....	521
12.6.1	建模 .....	472	13.10	环境设置 .....	522
12.6.2	材质与贴图 .....	472	13.11	渲染与后期 .....	523
12.6.3	灯光 .....	473	13.12	本章小结 .....	525
12.6.4	物理与逻辑 .....	474			
12.6.5	Python 程序代码 .....	474			

### 第三部分 实例篇

## 第13章 汽车建模与机械绑定 ... 477


13.1	准备与分析 .....	477
13.1.1	添加背景参考图 .....	477
13.1.2	分析建模思路 .....	479

## 第14章 角色模型制作 .....

14.1	造型与布线基础 .....	526
14.1.1	面部造型基础 .....	526
14.1.2	布线基础 .....	527
14.2	人物建模 .....	528



14.2.1 建立参考图 .....	528	15.3.3 设计场景材质 .....	586
14.2.2 面部建模 .....	529	15.4 灯光和镜头 .....	587
14.2.3 耳朵建模 .....	533	15.4.1 灯光参数 .....	588
14.2.4 细分优化 .....	534	15.4.2 镜头参数 .....	590
14.3 头骨建模 .....	535	15.4.3 设计场景灯光与镜头 .....	592
14.3.1 制作面部缺陷效果 .....	535	15.5 照明模式和渲染设置 .....	594
14.3.2 制作口腔结构 .....	538	15.5.1 照明模式 .....	594
14.3.3 制作眼球控制器 .....	540	15.5.2 照明的常规参数 .....	600
14.4 拆分 UV .....	542	15.5.3 场景渲染设置 .....	606
14.5 模型雕刻 .....	544	15.6 后期处理 .....	607
14.6 烘焙贴图 .....	547	15.7 本章小结 .....	608
14.7 映射绘制 .....	549		
14.8 毛发制作 .....	553	<b>第 16 章 角色动画</b> .....	610
14.9 着色器设计 .....	555	16.1 角色模型制作 .....	610
14.10 灯光与环境设置 .....	557	16.1.1 头部与身体建模 .....	610
14.10.1 三点式照明原理 .....	557	16.1.2 手臂建模 .....	611
14.10.2 设计场景灯光和环境 .....	557	16.1.3 腿部建模 .....	612
14.11 结点合成 .....	559	16.2 骨骼设计 .....	613
14.12 本章小结 .....	565	16.2.1 制作骨骼框架 .....	614
<b>第 15 章 室内效果图应用</b> .....	567	16.2.2 制作骨骼约束 .....	617
15.1 场景建模 .....	567	16.2.3 绑定模型与骨骼 .....	630
15.1.1 制作桌子 .....	567	16.3 制作步行动画 .....	631
15.1.2 制作椅子 .....	569	16.3.1 关键帧动画基础 .....	631
15.1.3 制作衣柜和相框 .....	572	16.3.2 制作关键帧动画 .....	632
15.1.4 制作地板和墙体 .....	574	16.4 制作非线性动画 .....	637
15.2 Yaf(a)ray 的安装与配置 .....	578	16.5 本章小结 .....	638
15.3 材质与贴图 .....	578	<b>附录 A Blender 大事记</b> .....	639
15.3.1 着色器 .....	579	<b>附录 B Blender 默认快捷键</b> .....	641
15.3.2 纹理贴图与通道映射 .....	584		



## Part1 第一部分

# 基础篇

Blender 本身的开发就是建立在和使用者密切沟通的基础之上，目的就是为了提高使用者的工作效率。与其他 3D 应用软件不同，Blender 的界面和用户体验十分特殊，熟悉其他 3D 软件的用户在刚开始接触 Blender 时，可能都会感到比较难以接受这种工作方式。但是一旦接受了这种模式，那么对今后的项目将颇有裨益。

另外，Blender 区别于其他 3D 软件的一点就是，它建立了很多全新的概念，例如引进了光标的概念、全新的物体和功能调用模式以及内部数据的 RNA 链接存储结构等。本篇将从用户界面和基本操作两方面开始，带领读者熟悉 Blender 的一些基础概念和操作方式，并分别从建模、修改器、灯光、材质和渲染等几个方面，简介使用 Blender 制作 3D 的入门知识和基本使用技巧。

- 第 1 章 Blender 概述
- 第 2 章 基本操作与界面
- 第 3 章 建模
- 第 4 章 修改器
- 第 5 章 灯光
- 第 6 章 材质与贴图
- 第 7 章 渲染



# 第 1 章

## Blender概述

### 1.1 开源简介

大多数的商业软件协议都规定，用户只被允许在单一的计算机上安装并运行该软件，并且不允许制作任何拷贝，同时用户也根本无法接触到源代码。但是如果是一款开源的自由软件，用户就完全不需要购买或者申请获取正版的授权。

#### 1.1.1 Blender 是什么

与一些十分昂贵、动辄数千美元的商业 3D 软件相比，Blender 是一款完全免费而且开源的 3D 制作软件。Blender 使用了 GNU 通用公共许可证（GPL）的授权，用户可以随意下载并可在多台计算机上运行，只需要同意并遵守自由软件基金会制定的开源协议即可。还可以下载 Blender 的源代码，对程序进行修改，将它应用于各种领域，也可以在改进程序后发布自己的版本，但是需要随版本提供一份 GPL 的拷贝，以保证程序接受者可以了解此协议下的权利。

作为一款应用软件，Blender 集成了用于制作各种 3D 内容的工具套件，包括建模、渲染、动画、后期处理、游戏制作以及跨平台 3D 交互制作等全方面的功能。因此对于不同的商业用户和个人艺术家，Blender 不仅可用于创建 3D 形象展示以及影视级视频，还可以结合实时 3D 引擎来创造可独立使用的三维游戏。

Blender 在过去 10 年中的发展十分迅速，它以强大的功能，加上开源和跨平台的性能，让全世界范围内越来越多的人都开始关注它，并成为它的用户。

#### 1.1.2 Blender 的发展与历史

1988 年，Ton Roosendaal 与他人合伙创立了动画工作室 NeoGeo。NeoGeo 很快便成为了荷兰最大的 3D 动画工作室，也是欧洲领先的动画工坊之一。在 NeoGeo 内部，Ton 任艺术总监负责内部软件的开发，经过仔细考虑后，Ton 认为当时 NeoGeo 使用的 inhouse 3D 工具过于陈旧，而且不易于维护和升级。于是在 1995 年时，Ton 带着他的开发团队开始重写这套系统，并且最终成就了我们今天所熟悉的 Blender。感谢你，Ton Roosendaal！

在 NeoGeo 继续完善和改进 Blender 时，Ton 认为可以向 NeoGeo 以外的其他艺术家发布 Blender，以便能获得更多的软件使用反馈。1998 年，Ton 决定成立一个名为 Not a Number (NaN) 的新公司，并作为 NeoGeo 的子公司用于拓展市场，以及维持 Blender 的后续开发。NaN 的商业模式主要包括：提供商业产品以及与 Blender 相关的服务，为公共计算机研究领域提供专业水平的动画和建模服务等。NaN 的核心目标就是要创建一个小巧而免费的跨平台 3D 制作套件，这在当时

绝对是一个革命性的观念! Ton 和他的工作室, 如图 1-1 所示。



图 1-1 Ton 和他的 Blender Institute 开源项目工作室

为了更广泛地推广 Blender, NaN 在 1999 年首次出席了 Siggraph 会议。Blender 的这次参与取得了巨大的成功, 大量的出版商和会议出席者都对它表示了强烈的兴趣。Blender 因此红极一时, 它的巨大潜力已经显现! 借助于这次 Siggraph 会议, NaN 在 2000 年早期便获得了 450 万欧元的风险投资。这笔投资使得 NaN 能迅速地扩展其业务, 不久便在世界范围内拥有了至少 50 名雇员, 为改进并增强 Blender 而工作。图 1-2 为 2009 年 Siggraph 上 Blender 的展台。

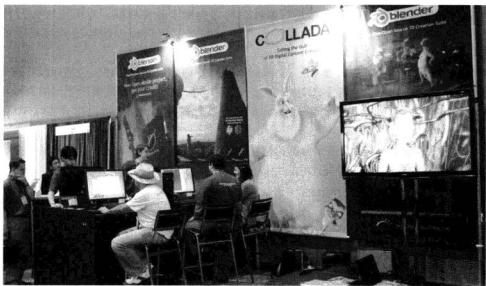


图 1-2 2009 年 Siggraph 上 Blender 的展台

2000 年夏天, Blender 发布了 2.0 版, 这个版本增加了一个集成的游戏引擎。到 2000 年年底, NaN 网站上的注册用户数量已经增加到了 25 万。2001 年 4 月, NaN 在新的投资商资助下又建立了一个更小规模的公司, 并在 6 个月之后推出了其第一个商业软件产品 Blender Publisher, 这一产品定位于独特的 Web 交互式 3D 媒体市场。但不幸的是, NaN 的雄心和机遇并不符合公司



的能力和当时的市场。由于产品令人失望的销售业绩以及持续艰难的经济形势，新的投资方决定停止一切对 NaN 的资助，这一决定也包括中断 Blender 的开发。

尽管此时的 Blender 还存在明显的缺陷，但它已拥有了一套完整的内部软件体系，以及尚不完善特性和一个用于提供 GUI 的非标准模式。考虑到来自用户团体和过去购买了 Blender Publisher 客户的热心支持，Ton 觉得不应该让 Blender 就此被逐渐遗忘。鉴于重新建立一个拥有足够数量开发团队的公司不太可行，于是 Ton 在 2002 年底决定成立一个非盈利组织——Blender 基金会。

Blender 基金会的主要目标是寻找一种可以继续开发，并且能促进 Blender 成为基于社区开源项目的方式。2002 年 7 月，Ton 设法使 NaN 投资者们同意这个单独的 Blender 基金会计划，以便为 Blender 开源。这一“自由的 Blender”之战致力于筹集 10 万欧元，则基金会可以从 NaN 的投资者们那里买下 Blender 源代码的所有权和知识产权，随后就可以将 Blender 发布给开源社区。让每一个人都感到惊喜的是，通过一群热情的志愿者（其中还有 NaN 的前雇员），基金会只在短短的 7 周时间里就达到了 10 万欧元的目标！

终于，在 2002 年 10 月 13 日，Blender 在 GNU 通用公共许可证（GPL）下向全世界正式发布了！Blender 的开发从这一天又开始了，推动这一项目的是来自广泛分布于全世界、全心全意的志愿者们，以及 Blender 的最初创造者——Ton Roosendaal。

2004 年，Blender 被首次应用于影视特效，《Spider Man II》是首部用 Blender 制作电脑动画部分的电影，如图 1-3 所示。评论界一致认为，与其他用昂贵的制作软件如 3ds MAX 制作的影片相比，这部电影的动作特效毫不逊色。

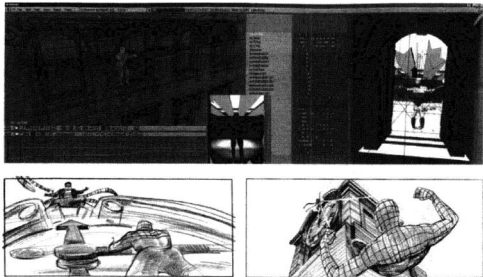


图 1-3 Blender 参与制作的电影《Spider Man II》

2006 年，世界上第一部开源动画电影《Elephants Dream》在耗时 3 年之后随新版 Blender 一同推出，并斩获大奖无数，该片也被评为当年 CG 行业最具贡献之作。2008 年，基金会又推出了第二部开源动画电影《Big Buck Bunny》，同步发行的新版 Blender 以其优化的粒子和毛发效果让人惊叹。Blender 也被评为当年十大 Animation/VFX 工具之一。2009 年上映的好莱坞电影《2012》更是应用了 Blender 中的 Bullet 套件来模拟电影中大部分的烟雾物理特效，获得了令人惊叹的效果，如图 1-4 所示。2010 年获得奥斯卡提名的动画长片《The Secret of Kells》，则使用 Blender 完成了部分 2D 制作，以及片中全部的 3D 特效。

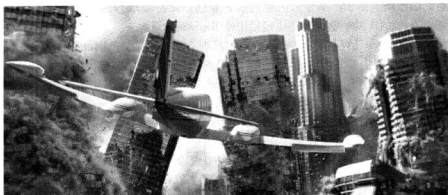


图 1-4 使用 Bullet 制作物理特效的电影《2012》

2009 年，新版本的 2.5 测试版正式推出，这次 Blender 再一次重写了软件的数据结构和用户界面等重要部分，并提供了更多的功能和改进的物理特效。2010 年 10 月，基金会的第三部开源动画电影《Sintel》正式上映，再次获得了巨大成功，同时改进后的 2.56 版也随电影同步推出。

### 1.1.3 社区支持

开源后的 Blender 在各国受到众多的追捧，并因此获得了迅猛的发展。它拥有全球最多的开发志愿者为其贡献代码和 Bug 维护。从 2002 年至今，Blender 源代码更新已超过 35000 次，更新统计示意图如图 1-5 所示。不仅仅是个人，Google 和微软等企业的开源研发部门也参与进来，并为 Blender 在 Windows 等多平台上的用户体验开发作出贡献。

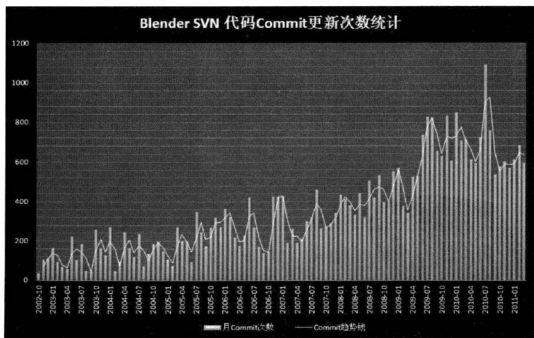


图 1-5 Blender 的 SVN 更新统计

从 1995 年至今，基金会平均每两个月都会发布一次小版本号的新版本升级，每次升级不仅包括了缺陷修复，还会不断地部署更多的新功能。如果希望查看每一次版本的更新细节，请参考官

方网站的版本发布记录。

作为开发者，可以登录 Blender Foundation Site 开发者社区。这里有所有正在开发中的功能项目列表，你可以直接从 SVN 服务器上下载全部的源代码和开发者文档。社区论坛还提供了开发中的常见问题解惑，以及 Python 脚本和 API 等文档更新，当然，论坛中还有供代码贡献者使用的头脑风暴讨论区。代码的开发和正式部署完全在 Blender 创始者 Ton 的监督下进行，核心开发团队会维护代码的质量和标准。Blender Foundation 也会定期参与 Google's Summer of Code 等项目计划，在那里众多年轻的开发者都会参与进来，为 Blender 奉献更多最新的功能代码。图 1-6 为 Blend SVN 主要代码贡献者示意图。

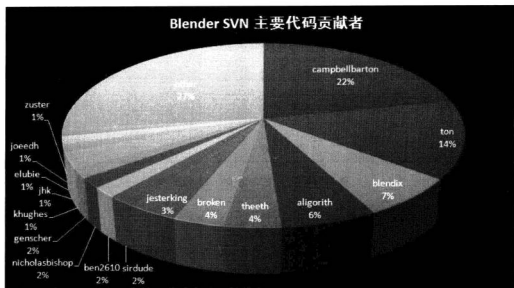


图 1-6 Blender SVN 主要代码贡献者示意图

不论是开发者还是使用者，都可以登录 Blender Artists 艺术家社区，在这里发布制作中的或已完成的作品，与来自全球各地的艺术家、游戏开发者和 Blender 爱好者们讨论交流。作为全球最大的 Blender 用户论坛，这里每天都会有上千条更新和回复，通过交流获得反馈是使参与者快速成长和进步的方法之一。在这还能找到大量丰富而全面的 Blender 免费教程和资源，这里人人都是老师，多参与讨论与提问、互相帮助、共同进步才是开源精神的真谛。

中文用户可以登录 BlenderCN 中文论坛，它是全球唯一认证的 Blender 官方中文论坛，这里提供了较全面的中文资料和汉化在线支持。

在线讨论也可以为你提供及时的信息反馈，登录 irc. feenode. net 上的 IRC 聊天室，选择进入感兴趣的 Blender 频道，即可与来自不同国家的 Blender 爱好者们一起参与在线交流，如图 1-7 所示，以下是聊天室的部分分类：

- ☐ #blender 可参与软件使用的相关讨论。
- ☐ #blenderqa 提供答疑。
- ☐ #gameblender 是为游戏开发者提供的交流平台。
- ☐ #blendercoders 是开发者的常用讨论区。
- ☐ #blenderpython 提供 Python 的 API 解疑和脚本开发交流。
- ☐ #blenderwiki 是 Wiki 志愿者的讨论区。
- ☐ #blendercn 是 Blender CN 中文在线讨论区。

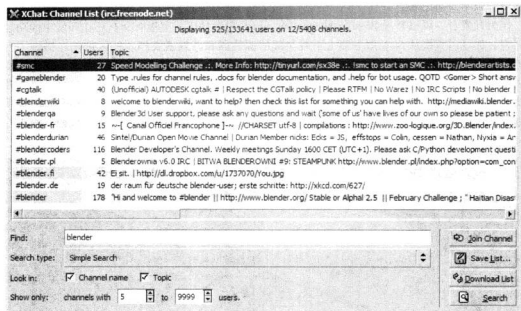


图 1-7 IRC 上的部分 Blender 频道

## 1.2 使用 Blender 的艺术家们

全球众多优秀的艺术家已使用 Blender 创作了很多优秀的经典作品，不少教育工作者也在不断为 Blender 提供大量教程和培训机会。在撰写本书的过程中，作者特意邀请了部分知名艺术家，请他们谈谈对 Blender 的认识，并为广大的读者解答许多新手入门时都会遇到的问题，以及一些关于他们与 Blender 之间的有趣故事。

### 1.2.1 Andrew Price

2009 年，Andrew 在澳大利亚创办了 BlenderGuru 网站，并提供免费的 Blender 在线教程。在短短的一年时间内，BlenderGuru 就实现了单日平均点击率超过 5000 次，并有 4000 多个邮箱订阅了其每日的更新，Andrew 在 Facebook 上也拥有一个 4000 多人的强大粉丝团，Twitter 上的追随者也超过了 1000 人，BlenderGuru 已成为了世界上最专业的 Blender 在线教育基地之一。因为他为 Blender 在线教育所做的贡献，他被邀请参加了 2009 年和 2010 年的 Blender 年会，发表了与培训和教育相关的重要演讲。2010 年 Andrew 撰写了 Blender 高级应用教程《The WOW FACTOR》系列丛书，并获得了极畅销的销量纪录，如图 1-8 所示。作为一名资深的培训师，Andrew 为我们写道：

大家经常会在网上看到一些类似这样的讨论：世界上最好的 3D 软件是哪一款呢？这真的是一个十分常见的争论，而且直到今天还有不少朋友在向我询问着同样的问题。我对这个问题的看法是：其实软件只是一门工具，只有艺术本身才能造就它的伟大！

不过大部分人在听完我的答复后，通常用两种方式来回我，要么抛来不屑的眼神，要么就干脆直接转身走开。但问题是，很多初学者都会步入这样的一个误区，那就是：如果我想要制作一款好莱坞级别的特效，即使这需要花费大量的资金，也一定要去配置一套最顶级的软件甚至硬件！但是事实真的是这样的吗？假如你将一支十分昂贵的画笔交给一位一点绘画经验都没有的人，你能指望他可以在一夜之间画出一幅像蒙娜丽莎这样的世界名作吗？显然这完全是不可能的！所以同样的道理，请你也不要将对作品的期望太多地依赖于一款软件的优劣上。

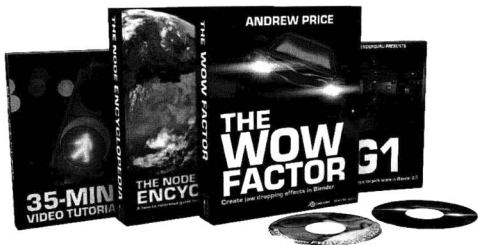


图 1-8 《The WOW FACTOR》

事实上，Blender 并没有完全进入工业级别的应用领域，这也确实是实话。但是这并不代表这款软件还不够优秀，而是因为大型的或者工作室的生产制作中，他们对软件的需求不仅仅是软件本身的功能，还需要相关功能的技术支持以及网络渲染等服务，而这些是开源免费的 Blender 还暂时无法提供的。但是大部分这些类似的功能和初学者都是不相关的，所以请相信我，如果你是一名刚入门的 3D 爱好者，那么 Blender 将足以提供和所有大型软件同样的基本功能和学习条件。

因此，在我看来，Blender 特别适合初学者。而且它的部分功能与一些大型软件相比，在某些方面还具有很强的竞争性，特别是工作流程和界面设计等最具特色的部分。还有更重要的一点，那就是它是完全免费的！如果你希望在结束基础知识学习后，转向学习并使用其他的软件，没有任何问题，因为 Blender 已经帮你完成了所有基本功的练习，使你可以很轻松地接受其他 3D 软件的学习和使用方式。

所以最后，我真心地希望你们能在不断地探索中去学好 Blender。可能刚开始你会遇到一些困难，但是只要你能有耐心并且坚持不懈，就一定可以渡过难关！加油！

图 1-9 为 Andrew 的代表作品《Remembering the Past》。



图 1-9 Remembering the Past

### 1.2.2 Anna Celarek

Anna 是一位年轻的波兰艺术家，现居住在美丽的维也纳。热爱艺术的她自学成才，早期一直使用 3ds Max 为主要工具，现在已完全改用 Blender 来制作她的所有作品。图 1-10、图 1-11、图 1-12 为 Anna 的部分代表作品。当我和她谈到为何会转向使用 Blender 时，她说道：

在讨论为什么我会喜欢上 Blender，并一直坚持使用它作为我的主要工具之前，我想先谈谈以前用过的一些其他软件。

我第一次接触数字艺术的时候才 15 岁，我学习的第一款 3D 软件是 Auto CAD。它可以让我随意地移动画面上的物体，并且还能执行一些基本的布尔运算操作，比起系统自带的绘图工具强多了，而且对于一个什么都不懂的小姑娘来说，这真的是太酷了！

刚开始的时候，我曾尝试去创造一些怪兽之类的角色，但是呢，AutoCAD 真的不是做这些玩意的理想工具，实际上它更适合那些工程师和建筑设计师们。可是我依然乐此不疲地去学习它，虽然当时还只能画出一些整脚的小动物、外星人或者飞龙等。我当时的想法就是，一定要掌握好这个软件，并且重新塑造这些小怪兽们！

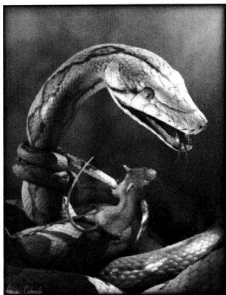


图 1-10 Snake and Mouse

不过接下来我发现了 3ds MAX，哇，它炫丽而丰富的功能简直太棒了，在和 AutoCAD 进行了简单对比后，我便全身心地投入到了 3ds MAX 的学习中。差不多在接下来的一年时间里，我都在学习网格建模法，这种方法给了我十分灵活的空间来创建生物体模型，甚至到现在我都一直保持着这种建模习惯。

当然这期间也有不少事情困扰着我，其中一件事就是我常常需要花费数小时的时间，去调整分解后的 UV 展开图，无数乱跳的顶点让我十分头疼。在寻找解决方法时，我幸运地发现了 Blender，它的解算能力真的很强大，同样的工作在 Blender 下只需要 3 秒钟就搞定了，我当时就震惊了！

在后来很长的一段时间中，我都只是借助 Blender 来帮助我完成 UV 的解算工作，直到有一天我决定开始尝试去制作我的第一部“大型”动画片（其实这个动画只有 1 分钟，但是对于初学者来说，这已经算是一个不小的项目了）。这部动画是为了参加当年由 CGTALK 举办的一届动画大赛，题目为“诡异的行为”，而我的想法是制作一部有关猫和老鼠的小动画。

但是在制作的过程中我又发现，在 3ds MAX 下制作骨骼和绘制权重真的很痛苦，不仅顶点在不停地乱跳，权重笔刷还难以控制，而且更糟糕的是镜像功能，它只能在猫咪身体上的一个小范围区域内有效，其余位置上完全不听使唤。我不知道错在哪，但是当这种煎熬达到忍无可忍的时候，我决定换一个软件试试。Blender 这次没有辜负我的希望，它成功地揽下了这项工作。Blender 的权重绘制十分灵活，镜像功能也相当高效，所有的操作都如此简单，困扰了我好几个月的骨骼问题只花费了几分钟就完全搞定了，而我仅仅是单击了 3 次鼠标！于是在剩下的动画制作中我决定完全采用 Blender，并且我又发现了它很多实用的功能，例如用于后期合成和材质制作的结点系统，以及集成的视频编辑系统。这时我才发现，Blender 真的很棒！



图 1-11 Cat in Tree

就在我制作这部动画的同时，Blender 又更新了好几个动画功能，例如改进的毛发梳理系统、更快更高质量的内置渲染器、SSS 着色器、全新的雕刻功能、快速的 AO 计算以及强大的骨骼系统。这其中我最钟爱的工具，当属 Grease Pencil 绘制工具了，它使我可以更轻松地在镜头视角下完成动作设计。

这就是我与 Blender 结缘的故事，我现在也一直在坚持使用它。与此同时我还发现，它的开发人员一直持续地专注于工具的改进和更新，而且我也从来没有见过哪一款软件能有如此快的更新速度。我想许多软件公司想的肯定都是赚钱，因此他们更愿意频繁地更新版本号来迫使用户付费升级，而不是去用心地更新软件本身。相比而言，Blender 的开发者却更专注于做好这个软件，使更多的新概念得以实现，让更多的实在功能可以得到部署。虽然有时觉得这么多的功能绑在一起有点混乱和繁杂，但是如此快的开发进度还真是让人感到十分兴奋。Blender 在各方面都在赶超其他 3D 软件，当然并不是全面胜出，可是我相信如果开发者能保持这样的更新速度，那么过几年后咱们走着瞧吧！



图 1-12 Freddie the Hen

从 2007 年底，开发者们就开始着手重写整个代码核心和用户界面，并命名为“2.5 计划”。不断推出的测试版也拥有了更多的新功能，并修复了大量的错误和缺陷。现在我已经开始使用新的 2.5 版来制作下一个新项目，讲的是一只大脚穿山甲的故事，相信不少的中国朋友都很熟悉

这种小可爱吧。即使现在使用的还只是测试版，但是我已经能感受到许多全新的强大功能，例如我可以直接使用第三方软件来制作贴图的 Project Paint 映射绘制，我用它完成了穿山甲的全部贴图制作，同时新的雕刻功能和间接光照也为人物和场景帮了不少忙。我期待着不久将要发布的更稳定的新版本！

对于刚接触 Blender 的朋友，我真心地祝福你们能和我一样，可以在学习 Blender 的过程中享受轻松和快乐，借用星球大战的名言：愿顺从的顶点与你同在！

### 1.2.3 Barath Endre

Barath 是一名优秀的匈牙利游戏设计师和图形工程师，他使用 Blender 为主要的游戏制作工具，代表游戏作品包括《Black Noise Engine》和《Ignite Racing Game》，同时他的个人动画《A Very Little Warrior》也荣获了 2010 年 Suzanne 最佳原创动画奖，如图 1-13 所示。



图 1-13 A Very Little Warrior

当我开始尝试学习 3D 的时候，用的是 3ds MAX 和 Maya 这些超级强大的专业软件，当然它们的个头也很大。但是在我不断地尝试和努力后，却还是没能学会，因为我没法用它们来理解 3D 的真谛。后来，在我接触了 Blender 之后，只花了 3 到 4 天的功夫，就能学会做一些简单的 3D 模型了，这给了我极大的鼓舞和成就感。

Blender 是一款逻辑性非常强的软件，比如建模，你可以直接对任何一个点线面结构进行非常直观的修改，而对比其他的软件，要实现同样快速的操作基本是不现实的，因为它们都太复杂了，光点线面的修改就需要调节大量的参数。而 Blender 则通过最简单的过程教会了我 3D 里面最重要的基础理论，那就是两点相连为一条直线，一点和一线能组成一个三角形。可能你们会觉得这个理论太初级了，但我确实是从学习这些基础理论开始一步步走过来的，而我当时也需要去理解这些最基础的东西，Blender 因此而成为了我学习 3D 的一根救命稻草。

### 1.2.4 Ben Dansie

Ben 是一位来自澳大利亚的数字艺术家，他专注于建模和材质设计。从能握住画笔的那一刻起，他便开始了绘画的学习，扎实而深厚的传统艺术功底给了他极强的 3D 领悟能力和表现力。现在他已经完全转向使用数字技术为主要的创作平台。Ben 还和另外两位艺术家 Jonathan Williamson 和 Jean Sebastian Guillemette 共同创建了 Montage Studio，用于提供免费的 Blender 教学视频。他们制作的教程现在已被大量转载于 CG Tuts、PACKT 和 CG Cookie 等知名的在线教育培训网站上。2010 年，Ben 受邀加入了 Durian 团队，参与了电影《Sintel》的材质与灯光制作。图 1-14 和图 1-15 为 Ben 的代表作品。



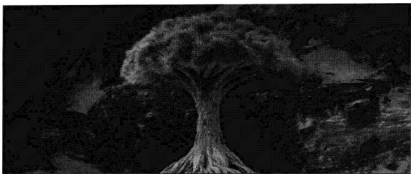


图 1-14 《Sintel》剧照：龙血之树

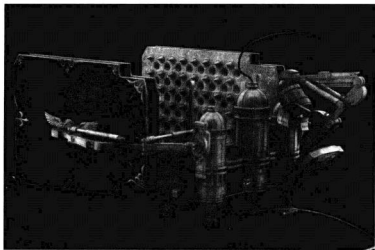


图 1-15 Steam Punk Mousetrap

接触 Blender 还是在我上高中的时候，那时只是在网上看了一些简单的教程，我也没能很好地去理解和学习它。几年后我进入了大学，有机会可以接触大量的 3D 制作软件，这时我决定好好地尝试并研究一下 Blender。很明显，作为一名在校的学生，它也是我当时唯一能够负担得起的软件，因为它是完全免费的。虽然其他软件有着相当丰富的教学资源，但是我很清楚，至少在毕业前，我可没有那么多钱去购买这些如此昂贵的商用软件。

后来我几乎使用 Blender 完成了大学中的所有课程设计，它也逐渐成为了我最主要使用的工具之一。当然在将它完全固定为我的主要工具之前，我也在不断地对比市面上的其他软件，了解它们各自的特性和未来的发展趋势，同时也在衡量着自己的经济承受能力。这时 Blender 基金会正式开始制作世界上第一部开源电影《Elephants Dream》，我兴奋地查看着官方 Blog 中每天更新的工作进度，并且期待着寄来的 DVD 和全套生产源文件。当电影正式发布后，我开始认真地学习这部电影的整个制作流程。感谢基金会发布的源文件，我很快就从主角人物 Prog 身上掌握了大量的材质、纹理、骨骼绑定和 UV 解算等宝贵知识。Blender 第一次让我体会到了开源与共享的学习优势。

从此我开始经常去登录 Blenderartists.org 论坛，与社区中的朋友做在线交流，这更是让我学到了不少课本中无法获取的知识。逐渐的，我也能开始帮助他人解决问题了，这种互动使我可以不断地获得更多的信息反馈，并巩固自己以前所学的知识。感谢 Blender 的这个强大在线社区，

它支撑了整个软件的进步和团队的发展，并且还吸引了大量的艺术家来这里分享和学习。同时，这样的在线讨论还为我带来了不少的自由职业收入，并且让我能有机会加入到 Durian 项目，参与电影《Sintel》的制作。

自由与开源精神不仅吸引了艺术家，还为 Blender 带来了全球范围的众多开发志愿者。这些来自不同国家的天才工程师们为 Blender 贡献了大量的创新功能，他们抛弃了隔阂，共同分享着计算机图形领域的最新技术。我将会一直坚持使用 Blender，同时我也希望本书能帮助你熟悉这款软件，解放你的思想，引领你共同踏进这片自由与创新的世界！

### 1.2.5 Colin Levy

美国小伙子 Colin 虽然今年才 22 岁，但是已经成功地执导了 Blender 的第 3 部开源电影《Sintel》，如图 1-16 所示。他还在上小学 5 年级的时候就迷上了摄影技术，13 岁时便制作了自己的第一部电影，在 10 年级的时候正式开始学习 After Effect 和 Blender 等 CG 特效软件，现就读于 Savannah College of Art and Design 的电视电影专业。



图 1-16 《Sintel》剧照：治疗

当我第一次打开 Blender 的时候，眼前的界面让我感到十分困惑。我花了好几个小时尝试着去做点什么，但是最后还是一头雾水，于是我决定直接退出程序，并将它从硬盘里完全删掉了。差不多在 6 个月之后，我才决定再去尝试一下。那时我已经浏览了许多优秀的 Blender 作品，想想如果那些家伙可以制作出这么精美的画面，我为什么不能？于是我耐心地重新开始，在认真学习了好几个星期的基础知识之后，我便深深地被这款功能强大的软件给迷住了！

当你能正视它的能力并熟悉它的各项功能之后，Blender 可以在你手中变得十分灵活而强大，你甚至可以用它来掌控一些复杂的大场景画面。但是作为一款时刻都在开发的软件，你必须要做好充分的准备去应对它随时做出的变化和更新，以及部分还未成熟的功能缺陷。幸运的是，对于《Sintel》这样的大型电影项目，我们能有一个开发团队可以全程配合我们的工作。Blender 基金会制作开源电影的优势就在于，他们的代码人员可以随时帮助我们解决制作中遇到的困难与挑战，以及各位艺术家针对电影提出的新功能需求。

我在工作中会经常使用 Blender，不仅仅是制作 3D 的静帧画面，也常用它来完成 2D 动画。作为一名导演，我特别喜欢使用它来制作影片的预览画面，辅助我去挑选镜头的角度，并且将故事板上的内容制作成 OpenGL 渲染的简易动画，演示给我的同事和演员。这为我们节约了大量的时间去理解整个拍摄计划，特别是制作一些拥有后期视觉特效计划的画面。对于我来说，Blender 特别适合完成这些类似的工作，因为它的操作十分灵活高效，允许我在片场就可以对很多画面做实时的调整。如果是拥有对话的画面，我则可以直接在 VSE 中剪辑整个对话，相当方便。

Blender 拥有一套十分强大的特效工具和合成系统，能够用于制作一些复杂的绚丽效果，我也常在自己的个人项目中应用这方面的工具。基金会的下一部电影项目被命名为“Mango”计

划，它将专注于实现更多的特效画面。但是如果要将 Blender 用于制作好莱坞级别的电影特效，那么它的 VFX 特效相关工具还需要更多改进。幸运的是，基金会已经在着手这方面的开发了！艺术家兼开发人员 Fracois Tarlier 为 Blender 列了一个未来的“特效计划”目录，而其中部分目标已经在新的 2.5 版中得以实现！

我相信你们在学习 Blender 之后，也一定能找到它最适合你工作的那一部分功能。我的下一部电影虽然使用的是真人实拍，但我已经借助 Blender 完成了所有的预览画面制作，并且计划在后期为影片添加一些特效。当然，我也更希望在未来可以再参与一部完全使用 Blender 的电影项目！

### 1.2.6 Enrico Valenza

Enrico 是一名来自意大利的自由职业插图设计师，同时也是一名 Blender 官方认证的培训师，他精通传统绘画，并长期为书籍杂志绘制插图和封面。2005 年他凭借动画《New Penguen》获得了 Suzanne 最佳动画和最佳原创剧本奖。2006 年，他加入了开源电影《Elephants Dream》的团队，2007 年又担任了开源电影《Big Buck Bunny》的美术指导。图 1-17 和图 1-18 为 Enrico 的代表作品。

我第一次打开 Blender 还是在它 2.23 版本的时候，那时它还没有正式开源呢。和大部分人接触它的第一感觉相似：“嗯？一个灰色布满网格的画面，一个窗口面板，还有一个怪异的方框和几个圆圈，这到底是个什么玩意儿啊？”我尝试去寻找一些能给我提供使用方法的提示信息，但是事实上它很难让我看到常规软件中一些相似的影子。最后我决定退出并删掉它，几个月后我几乎忘掉了它的存在。

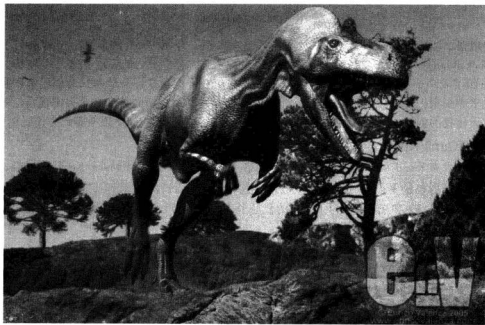


图 1-17 Allosaurus Fragilis

后来我决定再尝试的时候，无意中轻轻地滚动了一下鼠标中键，画面瞬间从普通的正交视图转换成了透视的立体视图，那几个怪异的方块和圆圈立刻转变成了一个立体的矩形和一盏灯光。嘿，这玩意看上去，好像也不是我想象中的那么神秘啦！

Blender 学习其实很轻松，只需要能参照一个完整的初级教程，就可以了解几乎所有的基础

知识。并且它“怪异”的操作界面也并没有为我带来任何困难和麻烦，我反而能从中感受到更多自定义和个性化的用户体验，这些是除了 Blender，在其他软件上完全无法体会到的。

今天，我们已经正式进入了 2.5 版的时代，而拥有更多令人期待功能的 2.6 也计划在 2011 年发布。为了淘汰落后的技术，开发者们在新的 2.5 版本中完全重写了全部核心代码，使得这款最初计划仅针对专业应用领域的 3D 软件，能持续拥有世界上最革新的技术以及最完善的功能。

在我看来，除了开源与创新，Blender 更伟大之处还在于，它始终在宣传一种积极向上的用户哲学，那就是它一直在努力为每一位用户都提供一个能踏进这门新兴数字艺术领域的机会，并创造了一个让人人都能去探索思维以及开发创新的公共应用平台。

作为一名自由职业的插画师，我已经将 Blender 完全作为工作中的主要工具软件很多年了，同时它还促成了我和多家意大利本土工作室的合作，因为现在已经有越来越多的团队开始接受并着手实施这条新兴却足够成熟的开源生产线了。

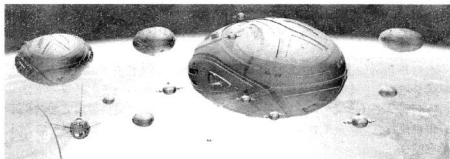


图 1-18 Golden Planet

我很荣幸能获得 Suzanne 动画大奖，并且还加入到了开源 Peach 团队中，全程参与了电影《Big Buck Bunny》的制作。对于我来说，这真的是一段相当令人难忘的宝贵人生经历。感谢 Blender 给了我一个成为数字艺术家的职业机会，它使一个像我这样，最初只会使用笔刷和画布的传统平面画家，现在却能如此快速地进入 3D 产业，并利用最新的数字技术去实现自己对幻想的追求，让我的创意从此可以在“四维空间”中，也就是时间和空间中以全新的方式展现！

### 1.2.7 Kamil Kuklo

波兰艺术家 Kamil 在 2007 年撰写了波兰的第一本 Blender 书籍《Blender. Kompendium》，并一举成为了当年的畅销书，同时也引发了波兰全国一股 Blender 学习的热潮。图 1-19 和图 1-20 为 Kamil 的代表作品。在谈到和 Blender 之间的故事时，Kamil 说道：

在我第一次接触 Blender 的 10 分钟内，数十个选项和下拉框几乎要让我绝望了，当然更谈不上任何 3D 图形学的用户体验。就这样我放弃了学习 Blender，转而去尝试着学习其他类型的 3D 软件。但是因为某些原因，最后我还是重新转回了 Blender，我想这也许是我人生中所做的最正确的一次重要决定吧。现在，我已经成为了一名能熟练使用 Blender 的 3D 艺术家，我有充分的信心告诉你，Blender 是世界上最好的 3D 软件！

我相信，通过这本书的学习，你可以很快掌握一套完整的项目制作流程。但是要想成为一名真正的 Blender 专家，还需要更多的耐心去多加练习。3D 创作的迷人之处在于它涉及了多种领



图 1-19 15 Minutes of Glory

域，比如建模、后期制作、动画或者雕刻等，幸运的是，借助 Blender 的强大学习平台，你可以很轻松地将它们全部掌握。当然，在你完成自己的角色或场景作品时，唯一真正制约你的并不是软件本身，而是你的创意和想象力。希望你们借助这款强大的软件，能够在 3D 的世界中更加酣畅淋漓地享受学习和创作！

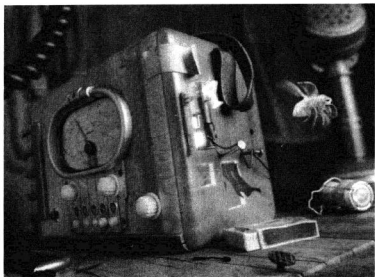


图 1-20 Nighthawks

### 1.2.8 Kevin Hays

美国艺术家 Kevin 使用 Blender 来完成他的所有雕刻作品，电影《Sintel》中的大龙就出自他之手，如图 1-21 所示。图 1-22 为 Kevin 的代表作品 Dragon Scale。对于 Blender 雕刻功能的学习，Kevin 向初学者提出了几点建议。



图 1-21 《Sintel》剧照：Dragon Scale

如果你是一位新接触 Blender 的用户，我的第一条建议就是不要刚开始就雄心勃勃地尝试去做一个大型项目。这样的项目，会很容易使你在还未真正体验到渲染、灯光和合成所带来的乐趣时，就放弃掉学习了。

相反地，我倒建议你你可以去尝试一个两三天就可以搞定的小项目。这样你不仅可以在十分轻松的学习过程中体验到整个制作的流程，还能享受实现最后成果所带来的成就感。当然我并不是想劝说大家不要尝试大型项目，只是对于初学者，特别是完全没有 3D 基础的朋友，从基础

的小项目开始才是学习的最佳途径。

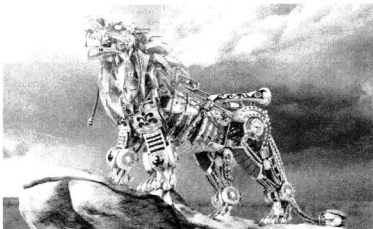


图 1-22 Steampunk Nemean Lion

如果你是一位新入行的数字艺术家，我的建议是尽量多地收集其他艺术家的代表作品，即使这些振奋人心的作品并不都是用 Blender 完成的。我个人就非常喜欢去 Zbrush 的论坛以及其他数字艺术网站的画廊。一旦发现那些能真正打动我的作品，我就会立即把它们下载添加到我的参考图文件夹里面。在以后制作自己的作品时，我都会先把它们翻出来作为参考资料。当然我们不可能直接去照搬这些作品，但是它们确实能为我在创作的过程中提供大量必要的解剖、光影、灯光和配色等相关参考信息。

### 1.2.9 Mathias Helmuth Pedersen

1990 年才出生的丹麦小伙 Mathias，却已经拥有了 5 年的自由职业艺术家生涯。他在 10 岁时便拥有了自己的第一台电脑，但并没有像同龄人一样去装上一大堆游戏，而是开始认真地学习数字艺术。通过不断的努力，他的 Blender 技术很快便在社区中得到了认可，并于 2007 年获得了 Blender World Cup 全球大赛冠军。图 1-23 和图 1-24 为 Mathias 的代表作品。在谈论 Blender 的同时，Mathias 还向初学者提出了学习中的 10 点建议。



图 1-23 Starry Night

Blender 是我最喜欢的软件，从刚开始接触到现在，都一直惊讶于这款开源软件的强大。我曾经错误地认为免费的玩意儿效果一般都不会太出众，但是 Blender 完全改变了我的这种观念。经过了几年的发展，今天 Blender 的强大已经让我感到十分震惊，感谢伟大的

的开源社区！



图 1-24 Tree of Life

当 5 年前我正式学习并使用 Blender 时，由于刚开始按照常规软件的学习思路，使得整个过程变得十分曲折和煎熬。但是当我开始逐渐掌握它之后，才意识它的不同之处，它带来的好处是使我的工作变得前所未有的高效！

小时候选择学习 Blender 还有个很无奈的因素，就是我在当时根本无法负担购买大型软件的费用，而今年我已经 19 岁了，依靠 Blender 我成为了一名全职自由职业艺术家，并且靠它来支付我的所有经济花销。Blender 不仅给了我实现创作 3D 梦想的机会，同时也给了我一个全新的职业生涯，没有它这一切都将无法成为可能。

今天的 Blender 已拥有了大量的在几年前只能称为是梦想的功能，当然我也期待着能在未来看到更多的新功能。不同于几年前，Blender 已经成为了一款真正的 3D 平台，同时开源社区中的志愿者还在不断地奉献并支持它的发展。我真心地希望你也能加入到这个团体，并成为其中的一员！

对于初学 Blender 的朋友，我希望能借此机会将自己总结出来的几点学习经验分享给你们。

### 1. 准备好你的求知欲望和好奇心

在正式开始学习之前，一定要充满对知识的渴望，并意欲主动地去发掘隐藏在 Blender 中的惊喜，而不是期待有谁会赐给你一本速成宝典。现在就立志开始学习吧！好奇心带来的动力能驱使你主动地学习，这甚至可以让啃掉上千页 Wiki 文档的过程也能变得轻松起来。找出所有令你感兴趣的领域，并以此为方向，然后再投入 100% 的热情，准备开始学习吧！

### 2. 多请教搜索引擎

这句话看似轻松，但是搜索引擎的确是你最好的老师，例如 Google 和 Yahoo 等。当你遇到某一个问题时，第一件事情就应该去登录它们并搜索问题的答案，同时尝试使用不同的关键词来查找结果。在大部分情况下，你的问题都可以通过搜索来解决，这比起向同伴发送求助邮件或者在论坛中发帖都要快得多，真的！

### 3. 尊重手册

Blender 的手册也就是官方提供的 Wiki 文档，可是很多人都会无视这套手册的存在，或者轻视它的重要性，并认为里面的内容对于一个初学者实在是太繁琐了。但是事实上，Wiki 手册也是 Blender 最重要的一个组成部分，社区和基金会都花费了大量的时间和人力在 Wiki 的编写和维护工作上。可能刚开始就要啃掉这么大块文档听上去很唬人，但是你只需要多花一点时间去认真地阅读它，就会发现里面其实蕴含着相当丰富而宝贵的学习资源。

#### 4. 论坛也能搜索

多去 Blender Artists Forums 等艺术家论坛上转转，上面拥有大量常见问题和对应解决方案的讨论，尝试着使用论坛工具栏中的搜索功能，去寻找和你类似的问题。你的问题肯定也会有其他初学者遇到过，特别是在新手区板块里，在这里搜索找到答案的几率也将会更大。尝试使用 AND 来组合多个关键词，这将会使你的搜索结果更准确。

#### 5. 给自己立下一个现实的目标

不要在刚开始就去尝试完成一个不切实际的项目，磨刀不误砍柴工，先耐心地巩固你的基础知识吧。最好是根据你目前的能力和水平，完成一个具有适当规模和难度的作品。大型却无法完成的工程学习，绝对顶不上一个小巧却可以成功实现的短期项目。要知道，独立完成一个真正的大型作品还需要很长的过程去学习和练习，所以请不要刚开始就眼高手低。

#### 6. 坦然面对批评和意见

把你的作品贴在论坛上，或者发在社区里，等待大家为你提出建设性的意见，这是促进你改进的又一个最佳方式。不要太溺爱你的这些早期作品，也不要因为大家的批评而灰心丧气，每个人的观点和视角都会不一样，这些恰好能从各个方面找出你现在的问题和缺点。当然也没必要为了这些评语就尝试去重新修改已经完成的作品，你要做的只是记住这些评论和观点，谦虚而坦然地接受所有的评价，然后在下一个项目中多加注意就是了。要知道，能有人真诚地指导你是一件非常令人感激的事情，而同时如果你能从这些评论中找到自己的问题，则更是一个十分令人兴奋的进步。

#### 7. 正视竞争

山外有山，人外有人，不要担心自己会落后于他人，也不要让这些去影响你的学习过程。嫉妒会让你感到难过甚至失落，你只需要享受你自己的学习过程，坚信自己的能力，真正的竞争对手其实只有你自己。

#### 8. 观察和学习

尝试着将你的视线从电脑屏幕上移开，回到真实的世界中，多去观察事物的本质，这能辅助你的创意与制作。万物是绚丽多姿的，它们都有着自己独特而美丽的一面。如果你想制作一支钢笔，那就去多观察一下真正的钢笔吧，或者在网上搜索一些参考图片。尝试以回归自然的方式进行创作，这能极大地提升你作品的艺术价值。

#### 9. 享受 Blender

如果你现在还没有从学习中得到快乐，那真的是件糟糕的事情。放松你的心情，在快乐中学习，在学习享受快乐，这样的心态可以让你事半功倍！

#### 10. 学会与他人分享

与他人分享你经验的过程，也是一种强化学习的过程。在这个过程中，你也会发现自己的知识在无形中得到了巩固和提升。当然，更重要的是这能够为他人提供很多帮助，让大家可以在你的经验提示中学习并获得领悟，就像我以上分享给你们的 10 点小经验一样。

### 1.2.10 Matt Ebb

来自澳大利亚的 Matt 是一名自由艺术家，他曾担任电影《Elephants Dream》的美术指导，负责完成了影片中大量的视觉特效制作。Matt 不仅是拥有深厚的艺术造诣的艺术家，同时也是一名优秀的软件程序工程师。2009 年，他被 Blender 基金会雇佣为全职研发人员，专注于 2.5 的内核开发和功能改进。图 1-25 和图 1-26 为 Matt 的代表作品。

刚开始接触 Blender 时，我还只是一名艺术设计专业的学生，那时我对 3D 制作十分感兴趣，并且希望能使用它来完成大学课程中的相关插图设计和网页项目。大学毕业后我正式成为了一



名视觉设计师。我也曾经使用过一些商业软件,但是相比较而言,Blender 是一款让我投资最少的软件,同时它特别适合像我这样的自由职业从业者,而且它完全足以胜任一些小型的专业项目制作。



图 1-25 Kajimba: Produced at Red Cartel ([www.redcartel.com.au](http://www.redcartel.com.au))



图 1-26 Lighthouse: Produced at Red Cartel ([www.redcartel.com.au](http://www.redcartel.com.au))

2005 年,我有幸能前往阿姆斯特丹,与 Blender 基金会合作完成了世界上的第一部开源电影《Elephants Dream》,这份工作经历极大地提升了我的 CG 技术,同时也鼓舞了我去全身心地投入到 CG 这门新兴的行业中。

我现在已经将 Blender 作为主要 CG 制作工具很多年了,让我坚持使用它的一个主要原因,就是在小型的团队合作中,Blender 可以让我很轻松地完成协作工作,并快速地解决项目需求,这一切都归功于 Blender 十分优秀的库管理模块。如果你需要实现这种类似的协作模式,我强烈建议你一定要去好好学习和利用这个功能。

另外一个原因就是,Blender 采用了一套全开源化的开发流程。首先,这提供了一个给你自定义开发的机会,当然前提是你需要知道如何去实现。当我越是热衷于 Blender 所带来的用户体验,就越有兴趣去尝试修改它的一些功能,并且希望将我在其他领域中期待的工具体添加进去。在

后来很长的一段时间里，我都自学着编程，并努力去改进 Blender 的内部工具，同时也开发了一些在我自己工作中所需要的新功能。

其次，Blender 相比其他软件的一个最大优势在于，你可以有机会参与到它的开发流程中，这和许多商业软件的封闭式开发模式完全不同。当然，和所有的软件一样，Blender 也并不是十全十美的，但是开源是它在这个领域中最独有的特性。加入到 Blender 开发的方式不仅仅包括提交代码，还可以向开发人员提出更多的功能建议，为新的版本提交及时的反馈以及测试报告，将你找到的 Bug 上传至跟踪服务器，以及为初学者创作教程和学习资源等。

而对于非开发人员，开源的优势也能为你带来更多的惊喜和帮助。我想对于很多人来说，能够亲手创作一幅 3D 艺术作品甚至一部小型动画片，都绝对是一件振奋人心的事情！开源的精神在于分享和奉献，作为一名艺术家，当你投入更多的精力去帮助整个社区时，反而能发现这其实能帮你更容易地熟悉软件中的所有功能细节。同时你在社区中的分享还能使你结识更多的朋友，你付出的越多，愿意为你提供回报和帮助的人也会越多。

在真正的专业项目制作中，我们常会遇到一些软件无法胜任的情况，并且希望能尽快获得可以解决对应需求的新功能，来帮助我们完成工作中的特殊应用。这时我们只需要在社区中去找合适的开发人员，并和他们及时地沟通，通常情况下问题都能在 1 到 2 天内得到解决。如果你曾经投入了大量的时间参与到社区的交流中，人们就会熟知你并会感激你在过去为大家所做的贡献，所以大家都会很慷慨地向你提供各种类似的帮助和支持。有时在你真正需要帮助的时候，社区真的是一条很实用的寻求帮助的途径！

无论是作为用户还是开发者，我都衷心地希望越来越多的中国朋友，能更积极地参与到这个逐步壮大的团体中来。在现在这个主要使用英文交流的社区中，我很少能看到中国社区成员的身影，但是我真的很期待。我相信借此机会，一定会有很多才华横溢的优秀中国艺术家和软件工程师能够了解 Blender，我特别希望能有更多的中文程序员可以加入我们，来帮助我们解决一些由于部分开发人员不熟悉中文而暂时无法完成的开发问题，例如改进系统对中文字符的支持等。

如果我能有幸为初学者提供一些建议的话，那么我希望你们能更积极地参与社区活动，这是一个可以让你结识更多朋友、接触更多新生事物的地方。我常看到不少初学者在社区中组织一些小型的在线比赛，例如速模竞赛等，这对于提高你的技巧、熟悉软件的操作是十分有帮助的。

当我希望去学习一门新的工具时，我会为自己设计一个从来没有尝试过的小型项目，让我可以在完全未知的情况下去实现探索性的学习。如果你对 Blender 中的某一个功能不是很熟悉，那么就去尝试做一个一定会使用到当前功能的小项目，努力去完成它，不要因为对某一部分技术不熟悉而轻易地放弃，坚持从探索的过程中去体会学习的魅力。

同时，不要在刚开始就雄心勃勃，也不要轻易地气馁。多去尝试一些新鲜的事物，用宽容的心态去面对自己以前犯下的错误，尝试将积累的经验融入到下一个项目中。我想如果你再回首自己早期的一些作品时，一定会想：如果让我再尝试一次的话，肯定会做得更好。如果你真的能这么想，那就说明你已经进步了！

最后，在你正式开始自己的 CG 漫游之前，我希望你能为自己摆正一个良好的心态。无论你是不是已经开始学习 Blender 了（我相信你一定正乐在其中），在这个长期的过程中，一定要意识到工具并不是最重要的，你的内心和你的作品才是学习过程中的重点。不要试图将学习封闭在单纯的理论和书本中，多去户外体验一下真实的世界，尝试一些非数字技术的艺术，例如摄影或传统绘画，这些都能为你打开眼界，并且激活你的艺术灵感。Blender 仅仅是一个工具，它在帮助你完成艺术创作的过程中，既能为你带来成就感，也会使你感受到挫折，因此不要太在意工

具在你创作中所扮演的角色和它所实现的目的，最后的作品是否能传达你所希望表达的构思和创意，那才是艺术的真谛。

### 1.2.11 Mike Pan

加拿大籍华人 Mike (潘辰) 从 2002 年起就开始学习并使用 Blender，他精通 BGE 游戏模块与实时特效开发，其作品已经被 4 本 Blender 书籍收录，同时还被作为演示程序在 Siggraph 大会上用于展示 Blender 的实时计算功能。他与 Dalai Felinto 合作的新书《Mastering Blender Game Engine》也将在 2011 年中旬正式出版。图 1-27 和图 1-28 是 Mike 的代表作品。

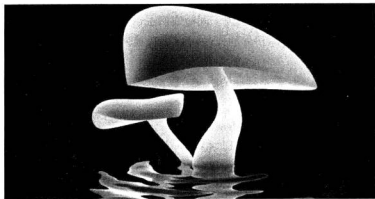


图 1-27 Mushroom



图 1-28 Real-time Marine Ecosystem Visualization for UBC Fisheries Centre

你可能曾经听说过 Blender 这款 3D 软件，或者你正在寻找一本教程可以帮助你更深入地了解 Blender，或者你就像我一样，在偶然发现这款软件之后，才意识到自己可以对某一事物产生如此强烈的激情与动力。当然无论如何，都欢迎你开始使用并学习 Blender！

新的 2.5 版基于开发者与用户多年的沟通和交流，已经使其逐渐演变为一款更加简单易学、具有更高的工作效率、同时还能提供更高质量应用性能的 3D 软件。你可以是一名为客户制作效果展示图的室内设计师，可以是一名电影或短片的动画师，也可以是一名寻求快速生产模式的游戏开发工程师，无论你的需求是什么，Blender 都能帮你快速地实现最终目标。

8 年前我开始学习 Blender 时，它还刚处于开源的初期，无论是学习资源还是相关的素材都

少得可怜，得益于在线社区对我的帮助和鼓舞，使我可以在短时间内就快速地获得成长和进步。随着时间的推移，Blender 的线上资源随着社区的壮大也在不断地增多，同时社区也在发展中得以稳固，并且依旧保持着热情友善的交流传统。

作为第一本在中国大陆发行的中文版 Blender 教程，本书绝对树立了中文 CG 行业中一块崭新的里程碑，同时这本书也紧紧跟随着 Blender 为整个行业带来最重要的一次变革。在这几年前，要想在个人电脑上实现免费的好莱坞级别动画制作平台是几乎无法想象的。但是现在，由于 Blender 所带来的技术和软件创新，整个行业的应用得到了平衡。使得今天许多艺术家，包括我自己，都能随意使用一些曾经做梦都无法想象的工具。基金会最新发行的电影《Sintel》也向世人证实了 Blender 在工业生产中的应用实力。同时我们也得感谢互联网为 Blender 发展所做出的推动贡献，高速的网络传播途径使你和你的作品可以很快被发送至世界上的任意一个角落，分享变得从未有过的简单。

希望本书能为有意踏入 CG 艺术行业的朋友，提供一条快捷的学习之路。

### 1.2.12 Pablo Vazquez

Pablo 来自阿根廷，是一名专注于 Blender 的自由职业艺术家，主持过开源游戏《Yo Frankie!》的开发，还出版了基金会的官方教程《Venom's Lab!》，并参与了电影《Sintel》的制作。同时他也是一名官方认证的 Blender 培训专家，在全球多个城市举办过 Blender 相关的技术研讨会。图 1-29 是 Pablo 的代表作品。



图 1-29 Frakas Dream!

今年差不多是我使用 Blender 的第 9 年了，它不仅从一款免费软件发展至今的自由开源，也同时逐渐融入了我的生活。因为它和用户具有很强的亲和力，无论是使用邮件还是在 Twitter 上参与讨论，任何一种方式都可以让你直接或间接地去辅助这款软件的改进。我所说的改进不仅包括对软件本身的代码维护，也包括资源类的更新，例如出版教程书籍或 DVD。我曾经花了很长的时间去适应这种新的思维，因为以前我始终认为大型的公司或团体一般都难以亲近，你也无法去期待能参与一款软件的发展和更新。但是 Blender 基金会和整个社区就像是一个小小的家庭，你不会再感受到封闭的隔阂，这里人人都很随和，互相聆听着彼此，善于沟通和交流。

那么我们应该从哪里开始学习和使用 Blender 呢？当然是你最感兴趣的部分啦！很多人都会先学建模，因为后面的学习都需要有一个基础的模型。我刚开始也是从一个最简单的模型开始，

做了一个“大屁股机器人”，没有去过分追求模型的细节，因为我不希望在刚开始就让前期的建模制作耽误我大量的时间。接下来我开始考虑学习材质和贴图等基础知识，在熟悉了基本流程后，我就进入了我最爱的动画学习。幸运的是，这时我得到了一份工作，虽然不是制作动画，但是却让我可以正式开始在真正的实践中去运用 Blender 了！

这是一部有关阿根廷的纪录片，我很庆幸通过这个项目熟悉了 Blender 的大部分基础知识，以及各类制作流程。这份宝贵的经验使我在接下来的 2008 年，可以有幸加入《Yo Frankie!》制作团队，并在 2009 年推出了我的第一部培训 DVD《Venom's Lab!》。在得到了 Ton 的认可后，他邀请我加入了《Sintel》电影的制作。同时在 Blender 开源工作室的这段时间里，我也有幸能在欧洲参加了大量的技术交流研讨会。现在我专注于渲染与特效，只在空余的时间里做一些角色设计等最基础的绘画工作。虽然我当前的主要工作是后期合成，但是我还是不希望让自己的绘画技术生疏，因为传统的美术功底可以让你保持对艺术的敏锐度，而技术只需要多加练习就可以了。

所以，学习是无止境的，同时学习也是不能停止的！充分利用你能找到的所有教育资源，在学习中不断地巩固自己，展示出你最好的作品，乐于帮助勤于奉献。不要偏向某一个特殊的知识，耐心地去控制每一个节点，享受单击 F12 后的乐趣，这就是最简单也是最有效的学习方法！

### 1.2.13 Robert J. Tiess

Robert 是一名优秀的抽象艺术家，他从 2003 年便开始使用 Blender 为主要工具，曾多次荣获 Suzanne 最佳角色动画和最佳数字艺术作品等奖项，他的作品以大胆的色彩和奇特的构思著称。图 1-30 和图 1-31 为其代表作品。

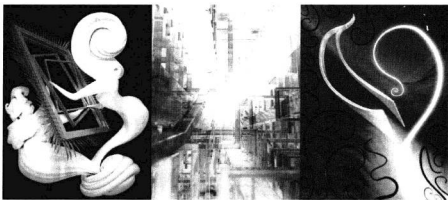


图 1-30 抽象系列



图 1-31 众神系列

Blender 是世界上最强大的艺术家工具之一！由于开发模式的限制，它的一些功能与其他软件相比还存在差距，但是功能上的限制并不是制约艺术家创作的主要因素，你的灵感和创意才是作品的真正灵魂。所以我在这里想说的是：不要过多地将自己的作品质量依赖于特定的软件上。

在决定自己专注于哪一款软件之前，我建议你可以先尝试一下 Blender，只要找一些简单的入门教程，感受一下它的优秀和独树一帜的地方，当亲身体会并对比之后，才能说服自己去做出明智的选择。我也是经过多次的选择和比较尝试之后，最终才将 Blender 确定为自己的创作工具的。

要想在艺术上获得快速的成长，你一定要主动地学习，而且最好从基础开始。我建议从最基础的传统绘画起步，并且多去研究知名艺术家作品的风格以及美术史。当你能熟悉并且可以辨识那些大师级作品的特色之后，才能逐渐总结并形成自己的独特风格。同时我建议你还得多尝试着去提升自己对于艺术灵感的捕捉能力，多使用全新的视角去观察世界，释放自己的思维！

当你能重新审视这个世界，并像艺术家一样去思考时，就会发现艺术其实可以让一切都变为可能，创意和灵感将无处不在！不要畏惧创新，不要给自己设定思维的定式和约束。尝试着使用不同的技术手段和多样的艺术风格，在这些全新的领域中去发现自己，并将自己的所思和所得展示出来。同时一定要多和大家交流，从互动中实现学习和进步，在探索中获得新的成长。

我们都可以成就自己的艺术家梦想，因为艺术本身就属于我们每一个人，艺术其实就是我们使用不同的视角去表现着同一个世界。学习 Blender 可以为你带来更多的全新体验，拥有去感受世界的另一个全新方式的机会，同时也可以帮助你去尝试超越自我，实现思维的突破。因为学习本身的含义就是去不断地发现，去体验和享受知识。这就是我多年使用 Blender 之后的体会，而且我还在继续学习，只因未来还未来，更多的未知还等待着我们的发掘和实现！

### 1.2.14 Sebastian König

德国艺术家 Sebastian 创办了 cmiVFX 在线 CG 教学网站，致力于提供高质量的 Blender 视频培训，从初级使用至高级合成应用，以及大量的 VFX 实践运用教学。图 1-32 和图 1-33 为其代表作。



图 1-32 Sebastian König for cmiVFX: Fly



图 1-33 Sebastian König for cmiVFX: Mammoth

作为一名经验丰富的 Blender 培训专家，Sebastian 说道：今年已经是我使用 Blender 的第 6 年

了，它可靠实用，一直都没有让我失望过。当初出于兴趣选择并学习它的时候，是为了完成我的大学毕业设计，但是后来我却逐渐迷上了这款软件。直到现在，它已经成为我的主要创作工具，而我现在的工作也正是一名使用 Blender 的自由职业艺术家，以及一名全职的 Blender 培训师。

人们常问我为什么会选择并一直坚持使用 Blender，而不是市面上常用的 Maya、Cinema 4D 以及 3ds MAX？我想大部分的 Blender 用户的回答可能都是“因为这玩意儿是免费的！”。当然这是它吸引大量用户的一个重要因素，而且如果没有开源模式，Blender 也许无法存活至今。但是对于我来说，它更重要的魅力还是它所带来的开源精神，以及快捷便利的操作模式和优秀的用户体验。

作为一名 3D 艺术家和自由职业者，软件是否开源或者免费并不能作为我选择它的主要标准，我的标准应该是它能不能帮我快速而高效地把任务完成，而且做出漂亮的结果并且令客户满意。花多少钱去购买软件对于我来说并不重要，毕竟为了工作，这点成本还是可以接受的。但是 Blender 对于我来说，它相当地高效、稳定，而且配备了足够强大的工具集，同时还拥有大量的开源资源，以及友好的社区和易于接触的开发人员。这么多的优势完全足以让我做出最终决定，选择它作为我的主要创作工具！

我很欣赏这种开源的精神和文化，为此我也创办了 cmiVFX 网站，提供类似的开源资源和具有工业标准的高级应用教学。在我看来，现在所谓的工业标准不仅有我们常讨论的 Houdini、Cinema 4D、Auto Desk 或者 Nuke，还应该包括 Blender 这颗未来的标准之星！人们曾将 Blender 看做一个简化的工具，或者仅供业余爱好者打发时间的玩具，但是事实已经告诉他们，这些想法早就过时了！

至少对于我来说，Blender 是一款完美的 3D 软件，它十分适合自由职业艺术家、小型的工作室或创业型团体使用，即使是在大型公司中，Blender 也可担任生产模式中的重要一员，它甚至还能够承担一条完整的生产流水线。Blender 拥有大量的闪光点，例如它的动画系统、建模方式、修改器模式、流体模拟、软体模拟、粒子和毛发系统、烟雾模拟以及内置的后期合成系统等，更重要的是，它每天还在不断地新增更多更新更实用的功能！

如果有 Blender 无法完成的工作，你完全可以借助于其他的软件，因为 Blender 提供了十分完美的接口功能。例如你要完成一个镜头追踪的特效，没问题，Blender 可以让你将镜头捕捉软件快速地嫁接到工作流的任意一个位置上；如果你觉得内置的渲染器还不过瘾，那么外置的渲染器包括 LuxRender、Octane 甚至 V-Ray 都可以随意由你调用，十分方便！

在大部分情况下，一个项目都不可能只使用一个软件来完成所有的工作，每一套工具都有它特有的强项。你有充分的理由和时间去做选择，但是我认为最重要的一点，就是它一定要是最高效的工具。Blender 的开源特性，以及它灵活的外置调用模式，快速的开发更新，都能使其成为兼容性最好的软件。我相信现在的 2.5 版已经能完全胜任你的所有工作，同时未来具有更多开发目标的 2.6 版更将值得我们期待！

### 1.2.15 Soenke Maeter

德国插画师 Soenke 可谓是 Blender 艺术圈中的大师级人物，他曾连续包揽了两届 F1 大赛冠军，和一届 Blender World Cup 全球竞赛冠军。2010 年，他加入了 Durian 团队，参与电影《Sintel》的后期合成制作，并获得了由 CGTALK 评选的 2010 年最佳杰出 CG 特效奖提名。图 1-34 和图 1-35 为其代表作品。

对于我来说，艺术的奇妙之处在于它可以帮助我们去看这个世界中的美丽事物。掌握一门软件，例如 Blender，并且不断地去追寻最新的 3D 技术，是用来塑造和重现你想法的必要手段。同时，永远不要放松对艺术认知的捕获力，毕竟我们都期待着你最终想要表达的作品结果。



图 1-34 Reaching Out

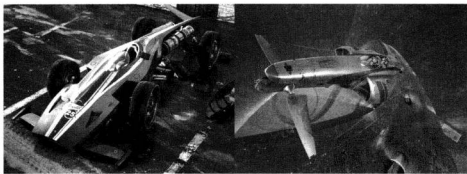


图 1-35 Blender F1 2008/2009

### 1.2.16 Tony Mullen

Tony 是世界上出版最多 Blender 书籍的作者，如图 1-36 和图 1-37 所示。他撰写的《Mastering Blender》，《Introducing Character Animation with Blender》，《Bounce, Tumble and Splash!: Simulating the Physical World with Blender》，《Blender Studio Project: Digital Movie Making》和《3D for iPhone Apps with Blender and SIO2》等书，多年蝉联亚马逊等网站的最佳畅销书排行榜。他在日本的 Tsuda 大学讲授 Blender 和 Python 相关课程，平时则担任独立制片人和动画师。当他获悉这本中文 Blender 图书的出版消息时，说道：

当我听说有一本即将面向中文市场的 Blender 书籍出版时，我十分兴奋。因为这标志着，Blender 在亚洲的推广和发展进入了一个历史性的时刻，Blender 的用户群将得到前所未有的壮



大，这对 Blender 和中文用户都是一件极大鼓舞的事情。Blender 对于亚洲市场还是一个全新的产品，不过我坚信，Blender 在这里将会很快得到快速的发展。

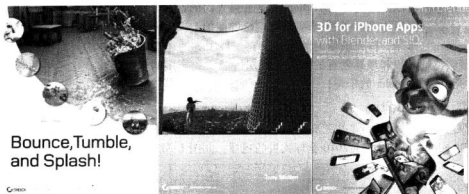


图 1-36 Tony 的 Blender 丛书



图 1-37 Tony 的 Blender 丛书

最近，一些来自亚洲的开发者也加入了 Blender 的全球化开发，这使得最新版本可以确保汉语和日语的国际化兼容，Blender 的用户能因此顺利地将其界面完全转换为本国语言，这对于本地化的推广有着十分重大的意义，感谢来自亚洲的开发者们！

我本人使用 Blender 已经多年，并长期在日本讲授 Blender 的相关课程。在担任讲师的这段时间里，我十分欣慰地看到，越来越多非常有激情的日本 Blender 用户正在不断地加入我们，甚至参与到 Blender 的维护与开发中。能够亲眼见证 Blender 在这个国家中的成长，是我最珍贵的人生经历。

Blender 为中国用户提供了大量的优秀功能。首先它本身就是一款自由软件，用户不需要考虑版权问题就可以随意拷贝和共享。学校、单位或相关培训机构可以非常快捷地建立一套基于 Blender 的 CG 培训平台，学生可以完全自由而免费地使用 Blender 做任何事情。对于工作室、自由职业者和众多的创意工作者来说，Blender 可以让你立刻开展工作，而无需要负担高昂的软件费用等初期创业成本。更重要的是，Blender 已经发展为一套顶级的 CG 创建软件工具集，它包含了建模、动画、物理效果以及游戏开发等相当丰富的功能。

另一方面，如果 Blender 能被为数众多的中国用户接受的话，这对软件本身是件好事，整个国际社区的建设也将受益匪浅。和所有开源软件一样，Blender 也需要依赖于庞大、活跃而且有建设性的社区支持。Blender 受助于公众的捐赠维持了软件的持续性开发，广泛的应用领域也使

得更多的艺术家和开发者开始使用 Blender，并同时贡献着代码更新和 Bug 修复。我希望能有更多的中国程序员可以加入我们，并投身于 Blender 的开发与更新，同时也期待更多中国艺术家创作出优秀作品，可以向全世界展示 Blender 这套强大的创意平台！

我希望本书可以为 Blender 带来更多更广泛的拥护者。Blender 是一套神奇的软件，无论是学习它还是使用它都是一件充满乐趣的事情。它拓展了你的思维，帮助你用最简单的方法去描绘出心中最美丽的幻想和最具创意的想法。我期待能看到更多的中文用户在这本书的帮助下，去体验学习和使用 Blender 的乐趣！

### 1.2.17 Vitor Balbio

巴西艺术家 Vitor 是一名技艺高超的 BGE 游戏设计师，他制作的《Lucy and the time machine》获得了 Blender Game Contest 2010 全球大赛的冠军，如图 1-38 所示。

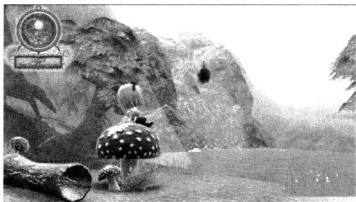


图 1-38 《Lucy and the time machine》游戏截图

曾经在很长的一段时间里，开源软件往往是低品质和业余的代名词。那时市场上的几款主流软件，在与低质量的开源工具进行了多年封闭式的竞争后，几乎完全垄断了整个市场，这使得行业中逐渐产生了一种排外的 3D 文化，使用户在选择软件时不是主要考虑它的实用性，而是考虑它在市场上的占有率。但是近几年来由于 Blender 的出现，正好改变了这种失衡的状态。

经过大量的比较与测试，Blender 已具备了实现高品质生产的标准和能力，完全可以实现各领域中的应用，包括数字艺术、3D 动画、广告和游戏制作等。

Blender 拥有十分全面的工具集、稳定的应用性能、完善的文档管理和非常活跃的全球社区。如果你是一位不满足于当前现有 3D 工具的技术爱好者，或者是专业的艺术家，我强烈建议你尝试一下 Blender。下载它，看看网上丰富的免费视频教程，参与到社区的讨论中，你将发现其实融入到这个全新的环境中很容易。这就是我接触 Blender 的亲身经历，也是现在更多专业艺术家正在经历的过程。你也准备开始并享受 Blender 带来的全新 3D 体验吧！

### 1.2.18 ДимитрКалинин (Dmitry Kalinin)

俄罗斯艺术家 Dmitry Kalinin 是一名自由职业艺术家，图 1-39 为其代表作品 From Another World。他说道：

我是在一年前才开始接触 Blender 的，那时完全出于一种好奇和兴趣，我决定在一个项目中去实验性地使用它，但是一旦当我上手之后才发现，它的能力远超出我的意料，而且非常容易掌握。所以现在，我已经完全使用它来制作我的全部作品了。

Blender 给了我一种动力去学习更多的新鲜事物，它让我见识了很多全新的概念技术和制作

方式，例如骨骼、渲染和合成。而且 Blender 还在不断地开发新功能，很多都是前所未有的，在这方面 Blender 绝对占据了技术的最前沿！

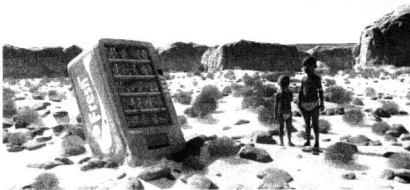


图 1-39 From Another World

以前我们可能需要同时使用多个软件，才能完成一个完整的项目。但是 Blender 的工具集相当全面，从刚开始的草图设计到最后的视频合成，你都可以找到对应的合适工具。同时 Blender 也拥有强大的兼容性，可以和市面上的一些商业软件完美地结合在一起工作，例如 LuxRender 和 Renderman 等。在格式转换上，Blender 也绝对是得心应手，你所知道的文件格式都可以在 Blender 上实现相互转换。所以在实际的工作中，我常使用 Blender 完成建模和雕刻工作，十分高效，而且质量也很不错。

在强大的开源力量之下，更多的艺术家已经开始使用 Blender 来完成他们的作品。我相信，未来的 Blender 一定会逐渐壮大，并成为主流应用中的一员。

### 1.2.19 黄海

黄海是国内最早接触 Blender 的代表用户之一，曾获得 2006 年 F1 大赛荷兰站第 2 名，如图 1-40 所示。他现为一名自由职业插画家。2007 年北京开源软件大会上，黄海代表 BlenderCN 做了题为《3D 开源软件 Blender 推介》的演讲。同时他也担任了国内多家知名艺术论坛的 Blender 专区版主，并长期在 BlenderCN 社区中使用网络 ID：DeathBlood 为大家提供大量的免费教程和学习资源，他为 Blender 在国内的发展和宣传立下了汗马功劳。图 1-41 为其代表作品。

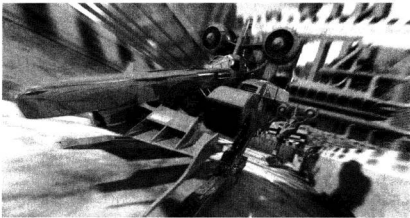


图 1-40 Blender F1 2006

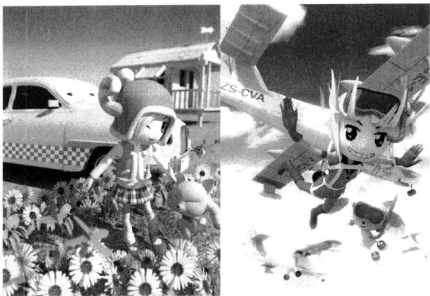


图 1-41 小女孩系列

几年前下载 Blender 完全是看中了它的娇小身材（当时只有 2M），并惊讶于其众多的功能，不过毕竟只是惊讶，随后就丢弃了！后来由于工作的变更，我需要一款小巧并且功能相对齐全的三维软件，脑海里第一且唯一的印象就是 Blender 了，建模、灯光、材质、渲染、动画，甚至还可以制作交互式 3D 和游戏，这些对于我的工作来说真的是太棒了！

在大部分工作中，我都会使用 Blender 来建模并创建材质，随后用 Yafaray 进行渲染以获得理想的效果。由于 Blender 自身是不支持 GI 渲染的，所以我也常选择外置的渲染器进行渲染，比如 Yafaray、LuxRender、VRay 和 Octane 等照片级别的渲染器。我的工作中也会经常遇到需要卡通效果的渲染，这一点在 Blender 下的制作效率相当的高。记得以前使用过 3ds Max 与 Finaltoon 的组合，但是它们的线条渲染相当费时，如果场景有一千万面，整个系统就几乎要崩溃了。而用 Blender 的 Edge 却完全没有这个顾虑，所以现在 Blender 已经成了我工作中必备的一个 3D 创作软件了。

Blender 这几年的发展相当迅猛（从它的体积就能看出来，已经有 20M 了），它也越来越能在各个领域胜任工作，如果你对大软件的价格望而却步，那么 Blender 绝对是不二之选。如果你初学 Blender，看着陌生的界面，请不要担心、退缩，当你真正走进后会就会发现，原来 Blender 不只是 Free（免费）也很 Free（自由），当你理解它的布局和基本功能之后，一定会慢慢喜欢上它的。

作为刚刚接触 Blender 的新人，肯定会遇到很多的难题，比较另类的界面、诡异的操作手法、无处寻找的工具等，这也都是我的亲身经历。早在几年前，我刚入手 Blender 时，简直就摸不着东南西北。所以我希望初学者首先应该多看看官方的 Wiki 文档，然后再跟着本书的介绍，慢慢地熟悉界面和所有菜单里的功能。当你对 Blender 可以达到了如指掌的时候，再认真研究本书中对项目流程的解析，我想这时你就一定能游刃有余地开始使用 Blender 创作自己的 CG 作品了。如果英文不是很好，也可以登录 Blender 的中国官方网站 BlenderCN.org，那里汇聚了很多国内的 Blender 使用者，大家一起学习的气氛更能给你带来鼓舞。

所以最后我想说的是，欢迎你进入 Blender 的大家庭！

## 1.2.20 王永海

王永海是国内知名的3ds MAX专家，曾著有《3ds MAX角色动画技术精粹：蒙皮·毛发·骨骼与绑定》，2006年开始使用Blender，并在BlenderCN社区中以Harrison为ID制作了大量的免费教程。2007年，由他制作的独立动画《Blood》荣获Suzanne最佳短片奖。他说道：

如果你买了这本书，说明你对这个软件感兴趣。当你开始学习和使用这个软件后，也许你会跟我一样开始对这个软件着迷，那么就先让我讲讲我跟Blender的一些故事吧。

### 1. 新欢

接触这个软件大概是4年前的事情了，当时我研究的还是3ds MAX。那时觉得自己在绑定和脚本这个领域上已经研究得差不多了，而软件也面临着本身所不能突破的局限，别的软件如Maya和XSI，都有很多比3ds MAX更适合做动画的优势功能。因此当时周围有一种氛围就是：我们该转而去学习另外一个软件了。

在此之前我已经花了很多时间去研究绑定和脚本，自己也做了很多教程与网友分享。当有人说这个软件不好的时候，我也会不自觉地进行抵触，也会经常不服气地说：这个用Max也能做啊！可是后来我渐渐地觉得，这些商业软件，我们花了很大的精力去学习，我还在一定程度上帮助推广过，结果到头来还根本没有能力去拥有一套属于自己的正版产品，感觉很不是滋味。当然我们在这里没有必要讨论版权的问题，所以我还是从软件选择的角度考虑好了。

当时我零零星星地看到一些关于Blender的内容介绍，于是乎也拿来研究了一下。那个时候的我对Blender的印象是操作怪异，界面也很简陋。但是我还是看到了一些新颖的吸引我的东西，例如骨架是一个物体，它的子层级才是骨骼，所以管理起来似乎很方便；构架很完整，粒子这样的东西虽然很简陋但也具备了；界面可以随意划分，你可以只显示当前工作最需要的部分。同时我发现，它的界面和操作还吸取了各个3D软件的特点，比方说层的管理来自Lightwave，父子关系的指定方式来自Maya (Ctrl + P)，当然应该也有XSI的一些特性在里面，后来还加入了MAX中很好用的Modifier功能。它支持二次开发，也就是支持脚本。最重要的是它还是一款开源的免费软件，也就是说你可以随意改动它的源代码，让它变成你自己所需要的样子。所以说，我也别去学什么3ds MAX的插件开发了（因为对我来说要进步还真要往这方面研究，有些对速度有要求的东西只能用脚本来实现），直接做一个自己的3D软件好了（不过我现在也没能达到这个地步）。

其实，Blender基本不用你动手，世界各地有很多程序员每天都在改进这个软件。一些大型的商业软件一年才出一个新版本，而且你也不要期望在它的新版本里面可以看到太多的新功能。但Blender则不是这样，我发现几乎每天都能看到这个软件的新改进。所以当时我决定选择学习的一个新的3D软件就是Blender。

### 2. 克服怪异的操作

刚开始学习Blender时，我的目标很明确：要把任何一个角落的功能都弄明白，于是我就把当时的用户手册和说明书全部打印出来学习。那时候的版本是2.31，软件的界面和操作风格在2.5之前基本上都保持着那个样子。

首先我要解决操作问题，因为那时还不能自定义操作，所以必须要适应那些操作和快捷键。作为一个MAX用户，可以说那是相当不习惯，除了视图操作，连选择操作在默认的情况下都是用右键的。很多功能都必须记忆快捷键，因为界面上既没有图标也没有菜单。

有时候我会埋怨为什么父子关系的链接不是先选择后拖拉呢？可是仔细一想，发现其实使用快捷键的方式不但速度更快，关键更准确！因为用鼠标可能要拉得很准才行，辛苦选择下来的多个物体有时候还会被误操作给取消。

另外为什么要用右键来选择呢？这是因为左键是用来做移动等操作的，如果还用来做选择，

可能会引起别的误操作（不过我后来还是改为习惯性的左键选择了）。

最后 Blender 的 GRS 等操作是按键盘后再随意移动鼠标的操作，而不是像一般软件用 QWER 来进行切换并拉动操纵手柄。其实你会发现这种方式远比点击那些手柄更自由，动画师应该喜欢这样更直观的操作。

关于操作我还是不多说了，因为进化已经融入到了 Blender 的发展历史中。你如果从别的软件转来学习 Blender，就一定会诟病这个问题。不过这一切矛盾到了 2.5 版本都已经烟消云散，因为现在你完全可以自定义快捷键了。

### 3. 共同成长

一开始，我觉得 Blender 真是个简陋的软件。不过无论什么软件，那些从一开始就一直坚持使用的老用户一般都“成精”了，所以我觉得很幸运（更早接触就好了），我在它简陋的时候开始学习它，这样一来，它每次改进和更新版本的时候，我们所需要学习的东西就很少了，并且可以很快掌握好新功能。

如果我们要更好地学习和理解 3D 技术，就有必要从 3D 的一些本源学起。也就是说，最好从我们做事情最不方便的时候开始，然后看着技术一步步的发展进步，这对于应用是很有效的。如果你捧起一本经典的《Digital Lighting & Rendering》，就会发现里面介绍各种灯光技术的进化顺序，和 Blender 中灯光功能的发展是完全一样的！例如：Shading 的类型，影子从 Buffer 到 Ray-trace 的计算改进，以前的 AO 和现在加入的光线反弹。

Blender 的功能发展也总是围绕着动画在进行，我们看到的这些基本上都是制作动画所需要用到的工具，而像 GI 那样还没有被动画渲染所普遍应用的技术，也还没有被加入进 Blender。

正如 Pixar 通过发展技术来满足自己制作动画的需要一样，Blender 用户的需求也被开源社区的程序员不断地满足着。在这个过程中，你不但看到了软件的成长，也见证了自己的成长。

### 4. 对缺陷的宽容

与别的软件比较，你总会发现 Blender 的一些缺点。我最耿耿于怀的是不支持多于 4 边的面（N-gon），其实准确地说是还没有针对多于四边面的细分功能，当然这个被称为 Bmesh 的功能已经在开发中了。有时候我们会说，这是个开源软件，别要求太高了。可是喜爱往往会带来期盼，这样的期盼是值得的。

我这里指的缺陷是除 BUG 以外的东西，有很多时候你问我某些效果在 Blender 里面能不能做，我只能摇头。其实，换个角度想，我们不能把所有东西都在一个软件中做完，虽然 Blender 有这个能力，但是有时候通过多软件的合作还是可以解决这些缺陷和矛盾的。比方说建模用 Silo，多边形建模方面就可以支持 N-gon 了；贴图方面用 3D-COAT，就可以支持图层了。这些软件虽然是商业的，但是相比之下它们的成本还是可以承受的，等于少换个手机，少买块用来玩游戏的显卡而已。

### 5. 不断有惊喜

到现在，我接触 Blender 已经有 4 年了，其中完全使用也有 2 年。它总是带给我不断的惊喜，很多 SIGGRAPH 上的技术论文，以及一些开放的新技术总是能够很快地被应用在 Blender 上。我让这些令我印象深刻的技术为例说明一下。

第一个是 AutoRig 或者叫自动蒙皮，这个功能是在你建好模后，再搭建一个骨骼，然后按下 Ctrl+P 选择自动蒙皮，Bang！你猜怎么样？蒙得比你自已刷的还好好！虽然手动更改有时也是需要的，但是对于我的工作来说，这一次按键解决了以往需要 1 至 2 天才能解决的问题。这是个秘密，我不大愿意告诉别人，因为这样别人就做得比我快了！还有一个秘密就是，在这个功能出来后，我常把模型从 3ds MAX 导出到 Blender，待完成蒙皮后再导回去（反正那些强大软件的拥护者大多不会看这本书，所以肯定不会知道这个秘密的）。

第二个是 Ptex，现在已经有人集成到 Blender 中来了，这样以后就再也不需要分 UV 了。感谢迪士尼，估计这个功能在这书出版后不久就会成为正式的版本了。

第三个就是现在还在开发的自动雕刻细分，也就是 Sculpttris，这个功能可以说相当具有革命性，要不那个软件也不会被收购了。

一般来说很多程序员都会把 Blender 作为一个基本平台，用来将那些最新的 CG 技术付诸实施。我们可以很轻易地下载到这些不同分支的特殊版本，另外 Google 也参与了进来并支持很多相关项目的开发。

除了新技术，修正 Bug 的速度也是很让人惊讶的，Blender 的程序员都很勤奋，我提交的 Bug 报告很多都能在 1 到 2 天内得到解决。我很享受这个过程，那种心态跟消费者享受售后服务不同，因为你会感觉到自己是在帮助开发人员。时间久了，你还会发现，这个软件是在按照你期望的方向发展。其实这是社区的人群策群力共同地在改进某些功能，久而久之，用户的需求就成为了具体的功能实现。

## 6. 远离宗教似的狂热

因为 Blender 就像 OS 里的 Linux，很多爱好者就会打类似 Windows VS Linux 之类的口水仗。我喜爱这个软件，但那只是一个工具，使用者的水平才是更重要的。我们希望它变得更好，变得比别的软件好，同时我们也要提高自己的能力。我不会排斥别的同类软件，但对于一个软件来说，能吸取别人的优点是最重要的。就像操作方式，只有变成大家都习惯的方式，或者说能够自己定制才是最佳的方式。一方面人要适应软件，另一方面软件发展也要人性化。

我们不需要一些为少数精英打造的高级玩具，我们要的是大众都能掌握的好软件。特别像动画制作往往是团队合作，如果是一个很难上手的软件，那么培训起来也会很费时。我想起以前经常会有 Maya VS MAX 这样的讨论，实际上这些 3D 软件覆盖的功能范围都很广，整体比较意义不大，你可能只能取某个功能来具体讨论。

可笑的是当时我们都没有意识到，我们维护的不过是手里一个没有名分的盗版，而且原来的三家现在已经属于同一个主人了，所以有很多人因为不想被软件公司“劫持”转而使用 Blender。如果要再遇到各种 Blender VS XXX 的情况，我也用不着太过维护，因为它是一个争气的“孩子”，它已经实现了很多我的愿望。

我所提倡的就是使用开源软件来制作独立的动画，这样的门槛低，适合爱好者。Blender 当前所提供给我们的功能已经能完成相当高质量的动画了，所以为什么不用它去做一个自己的东西呢？

关于 Blender 想说的实在太多了。最后，我想告诉读者朋友们，你所面临的那个问题：是否对 Blender 说 Yes，这是一个选择问题。我发现有时候人是很难被说服的，我也不大喜欢去说服别人，但是我发现自己的一些不经意的行为却有可能会对他人造成影响。以上文字只在于让你能从另一个人的体验中先去了解这个软件，然后再做出自己的选择。

## 1.3 Blender 的安装与配置

### 1.3.1 系统要求

#### 1.3.1.1 硬件环境配置

对比 Maya 和 3ds MAX 等大型 3ds 制作软件，Blender 不需要很强的计算机硬件配置，甚至可以在 iPhone 上运行它<sup>①</sup>。Blender Foundation 提出的个人电脑最低配置要求如下：

□ 三键滚轮鼠标

① 当然，配置越高的硬件能提高 Blender 的工作效率。

- ☐ 300 MHz 的 CPU
- ☐ 最低 128 MB 内存
- ☐ 具有 1024 × 768 分辨率的 16 位显示器
- ☐ 至少 16MB 显存的 OpenGL 图形卡
- ☐ 最低 20 MB 硬盘空余空间

当然为了获得更好的性能，这里推荐你的配置至少是：

- ☐ 三键滚轮鼠标，推荐配合使用 WACOM 等数位板
- ☐ 多线程的 64 位 CPU
- ☐ 8 GB 内存
- ☐ 具有 1920 × 1200 分辨率的 24 位以上双屏显示器
- ☐ 768 MB 以上显存的 OpenGL 图形加速卡，例如 ATI Fire GL 或者 Nvidia Quadro

Blender 支持 32 位和 64 位硬件系统，在 64 位 Unix 系统下无 2 G 最大内存限制。渲染支持多线程 CPU，例如 Intel 的 Core i7 和 AMD 的 X6 芯片组。Blender 支持各类手绘板系统，并可兼容在 Windows、OSX 和 Linux 等操作系统下使用。

### 1.3.1.2 操作系统与环境变量配置

Blender 可以运行在任何操作系统上，以下是 Blender 支持的常用系统列表：

- ☐ Windows 98、ME、2000、XP、Vista 和 Win7
- ☐ Mac OS X 10.3 及最新版本或 PowerPC
- ☐ Linux i386, x86\_64/AMD64
- ☐ FreeBSD 6.2 i386 Irix 6.5 mips3
- ☐ Solaris 2.10/Sparc

Blender 提供了两种安装方式，一种是利用程序解包来安装，另一种是源代码自行编译。可以进入 Blender 官方主页的下载区<sup>①</sup>，点击 Downloads 下载对应系统的安装文件。

Blender 的二进制安装文件共有如下 4 个版本：

- ☐ Windows 32/64 bit
- ☐ Linux x86-32/64
- ☐ Mac OS X-PowerPC/Intel
- ☐ FreeBSD

其中，Linux 版又分为 4 个不同版本：Intel 架构版、PowerPC 架构版、静态链接库版和动态链接库版。区别动态链接库版和静态链接库版很重要，因为静态版在编译时已经包含了 OpenGL 的库，从而使 Blender 可以在显卡无硬件加速的系统上运行。如果你的机器上不需要使用静态版，那就请使用动态版。

在 Blender 里 OpenGL 被用于各种图形的绘制，包括菜单和按钮。因此，要执行界面操作的系统必须安装 OpenGL。对 OpenGL 的依赖要求安装的 OpenGL 必须是正确而且是兼容的，要知道，不是所有的 3D 显卡生产商都能提供这样的兼容性，尤其是那些游戏中的廉价显卡生产商们。

Blender 使用了 Python 作为脚本支持，默认安装的目录中会包含一些基本的库文件，但是对于一些特殊的编译版本，以及在 Linux 系统下运行完整的脚本程序，必须安装 Python 的完整程序库。

登录 Python 官方网站的下载专区<sup>②</sup>，点击页面上的 Downloads 下载即可。安装完毕后建议对环境变量做一些设置，将 PYTHONPATH 路径指向 Blender 的脚本文件目录。例如在 Windows 系统

① 网站地址：<http://www.blender.org/download/get-blender/>。

② 网站地址：<http://www.python.org/download/>。



下采用了默认路径安装,可在环境变量中设置 Blender 的默认脚本位置 PYTHONPATH 为“C:\Program Files\Blender Foundation\Blender\scripts\bpymodules”。当在启动 Blender 时,如果能在控制台窗口中看到如下打印输出,就说明已经正确地安装好 Python 了。

```
Compiled with Python version 2.6.6.
Checking for installed Python... got it!
```

### 1.3.2 快速安装

针对 Windows 系统,双击下载的可执行二进制文件 blender-2.##-windows.exe,即可开始快速安装。单击接受证书等条款后继续安装,完毕后可以在桌面和开始菜单中找到可执行的 Blender 主程序,或者直接双击\*.blend 项目文件运行主程序。

若是 OSX 系统,可双击解压安装文件 blender-2.##-OSX-10.3-py2. #-ARCH.zip,在目录下即可看到主程序文件,双击图标就可以运行 Blender 了。因为 Blender 使用 OpenGL 来渲染用户界面,而 Mac OSX 也使用它来绘制整个桌面,所以你应该保证系统有足够的显存,Blender 在显存小于 8 MB 时无法运行。为了达到 16 MB 显存,可以将显示设置成“1000s of colors”(System Preferences -> Displays)。

在 Linux 系统下,需要下载一个适合自己机器的版本进行安装,下载专区提供的文件名为 blender-2.##-linux-glibc#. #-ARCH.tar.gz,这里的 2.## 是版本号,#. #-# 表示为 glibc 版,而 ARCH 则表示机器构架,类似的有 i386 或 PowerPC。将其解压到所选择的目录中,在其中可找到 Blender 的执行文件。或者首先运行 X 图形系统,打开 Shell 执行 ./blender 即可。

### 1.3.3 本地编译

#### 1.3.3.1 源码下载

本地编译可以使用 Visual Studio 2008,也可以使用 MinGW,同时还需要下载 Cmake 和 Tortoise SVN,来帮助完成编译环境的部署。当软件安装完毕后,就可以开始下载 Blender 源码了。首先使用 Tortoise SVN 从以下服务器下载最新版本号的源码到本地。

```
https://svn.blender.org/svnroot/bf-blender/trunk/blender
```

这里将源码下载到目录 C:\blender\blender,接下来再下载编译所需要的库文件,使用 Tortoise SVN 从下面的地址中下载即可。

```
https://svn.blender.org/svnroot/bf-blender/trunk/lib/windows
```

将库文件放置在 C:\blender\lib\windows 即可,如果需要编译 64 位的 Blender,那么需要下载如下库文件,并放置到 C:\blender\lib\win64。

```
https://svn.blender.org/svnroot/bf-blender/trunk/lib/win64
```

#### 1.3.3.2 配置编译

使用 MinGW 执行编译时,需要添加如下脚本来自定义编译配置,首先新建一个配置文件,加入如下代码:

```
#C:\blender\user-config.py
WITH_BF_FFMPEG = True
WITH_BF_OPENAL = True
WITH_BF_GAMEENGINE = True
WITH_BF_PLAYER = True
WITH_BF_JACK = True
WITH_BF_SDL = True
WITH_BF_ICONV = True
WITH_BF_COLLADA = True
WITH_BUILDINFO = True
```

```
BF_DEBUG=False
BF_NUMJOBS=1
```

如果计划编译 64 位 Blender，则需要取消 SNDFILE、JACK 和 QUICKTIME 等选项，因为这里根本就没有针对这些功能的编译库，修改脚本文件如下：

```
#C:\blender\user-config.py
WITH_BF_FFMPEG=True
WITH_BF_OPENAL=True
WITH_BF_QUICKTIME=False#不支持 64 位编译模式
WITH_BF_GAMEENGINE=True
WITH_BF_PLAYER=True
WITH_BF_JACK=False#不支持 64 位编译模式
WITH_BF_SDL=True
WITH_BF_ICONV=True
WITH_BF_COLLADA=True
WITH_BF_SNDFILE=False#不支持 64 位编译模式
WITH_BUILDINFO=True
BF_DEBUG=False
BF_NUMJOBS=1
```

如果使用 MSVC 来执行编译，则不需要使用脚本来执行编译配置，只需要使用 CMake 打包制作“Solution”文件，并导入到 Visual Studio 中进行编译即可。

首先运行 CMake，在顶部的源码目录中，输入放置 Blender 源代码的位置，这里为 C:\blender\blender，如图 1-42 所示。接下来在下面的编译目录中选择打算存放编译文件的目录，例如 C:\blender\build。最后单击 Configure 按钮，在弹出的对话框中选择编译器即可。

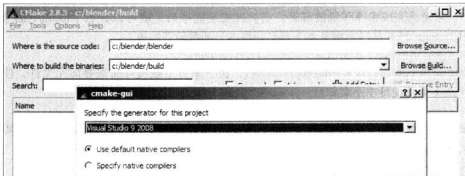


图 1-42 运行 CMake

单击选择 Visual Studio 后确认，CMake 会弹出一个列表，供选择编译配置，在 32 位 Blender 的编译模式下，可以保持默认选择。一旦完成了这项设置，CMake 会在指定目录下生成一个 Solution 文件编译包“Blender.sln”。

#### 1.3.3.3 运行编译

最后可以关掉 CMake，开启 Visual Studio 2008，在刚才的路径（C:\blender\build）下打开编译包文件“Blender.sln”。将默认的编译目标从 Debug 转换至 Release，如图 1-43 所示。并选择编译菜单中的 Build Solution 运行工程，即可以开始编译了。

大约 10 分钟后，就可以在 C:\blender\build\bin\release 目录下找到刚才编译的可运行 Blender 程序了。

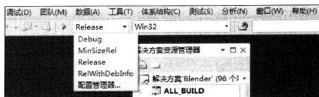


图 1-43 Visual Studio 下执行编译输出

如果使用 MinGW 来完成编译,则需要首先新建一个后缀名为 .bat 的批处理文件,并在其中添加如下代码:

```
set PATH=% PATH% ;C:\Python26;C:\MinGW\bin;C:\MinGW\lib
```

```
cd C:\blender\
```

```
python scons\scons.py BF_TOOLSET =mingw
```

其中,Python 的路径为安装路径。单击执行这个批处理文件,系统将自动调用 MinGW 执行编译操作,最后的编译输出可以在目录 C:\blender\install\中找到。

#### 1.3.3.4 更新 SVN

如果需要重新编译至更新的版本,需要首先对编译环境执行清理操作,将旧的文件从编译环境中去掉,在控制台中使用如下代码:

```
python scons\scons.py -c
```

接下来如果需要从 SVN 服务器上更新 Blender 代码,只需要在路径 C:\Blender\下创建一个批处理文件,并添加如下代码,运行即可:

```
cd lib\win32
```

```
svn update
```

```
cd .. \blender
```

```
svn update
```

```
pause
```

如果是 64 位的编译环境,则修改代码如下:

```
cd lib\win64
```

```
svn update
```

```
cd .. \blender
```

```
svn update
```

```
pause
```



## 第 2 章

# 基本操作与界面

Blender 采用了一套全新的用户体验机制，它的系统界面和操纵控制方式与 Windows 下的其他同类 3D 软件完全不同，这种差异性也是 Blender 强大功能的基础。在本章中，我们将学习 Blender 中的一些基本概念，目标是使读者能熟悉软件的视图与界面，并熟练掌握一些基本的控制与操作方式。

## 2.1 Blender 基础概念与基本操作

### 2.1.1 鼠标操作与键盘常用功能键

在本书中，将鼠标的左键单击操作统称为 LMB（Left Mouse Button），鼠标的滚轮操作简称为 MMB（Middle Mouse Button），鼠标的右键单击操作则简称为 RMB（Right Mouse Button）。

官方推荐在使用 Blender 的过程中选择三键的滚轮鼠标，如果使用的是双键鼠标，也可以借助组合键 Alt + LMB 来替代滚轮功能。针对 Mac 苹果电脑的单键鼠标用户，可以使用 MB（Mouse Button）来替换 LMB 操作，用 Option/Alt + MB 来代替 MMB 功能，使用 Command/Apple + MB 来替代 RMB 功能。

同时，请尽量选择标准的 Windows 全键盘，如果使用的是没有小键盘的笔记本，也可利用组合键 FN + Num Lk 来开启数字小键盘功能，或者在参数设置中激活 Emulate Numpad 选项，使用数字键 0~9 来代替小键盘。

本书中的所有快捷键范例，均使用大写字母来表示，组合键操作则使用“X + X”来表示。例如，“G + Z”表示先单击快捷键 G，再单击快捷键 Z。小键盘的数字键分别使用 Num0~9 来表示，其他的常用快捷键还包括 Ctrl、Alt、ESC 和 F1~F12 等。

### 2.1.2 物体与原心

在 Blender 中，一个基础的控制单位称做 Object（物体）或者对象。一个物体可以是一个网格模型，也可以是一盏灯或是一台摄像机。每一个物体都有一个 Origin（原心），用于标识物体本地坐标系的原点和控制杆的默认位置。如图 2-1 所示的多边形、灯光和摄像机上的橙色圆点，分别为各个物体的原心。

在 Blender 中对物体做选择操作，是将鼠标的光标移动至物体轮廓线内的任意表面位置上，然后单击鼠标的 RMB 即可。如果需要同时选择多个物体，可以使用组合键 Shift + RMB 来完成多选操作。当一个物体被选中时，它的轮廓线会默认变成橙色，此时其原心位置也会出现一个控制器，分别使用了红、绿、蓝三种颜色，来表示 X、Y 和 Z 轴三个坐标方向以及对应方向上的控制杆。在控制杆的箭头处单击鼠标 LMB，即可对控制杆执行相应的移动或旋转等

控制操作。

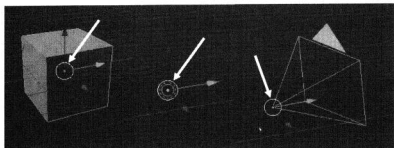


图 2-1 不同物体的原心

## 2.1.3 Blender 的界面操作

### 2.1.3.1 界面 Layout (布局)

图 2-2 为 Blender 的默认启动界面，界面中的窗口按照功能进行区域划分。

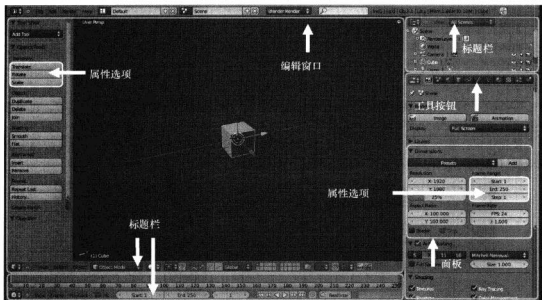


图 2-2 Blender 的默认窗口布局

视图编辑窗口是 Blender 的主要窗口之一，用于显示当前 3D 场景中的所有物体，包括灯光、摄像机和多边形物体等。面板窗口用于分类列举所在工具面板中的控制面板，其中的属性选项包含了 Blender 所有工具和对应工具的功能属性参数。标题栏显示了当前的窗口属性，可以单击上面的负号箭头按钮隐藏标题栏。

### 2.1.3.2 新建 Layout (布局) 与 Scene (场景)

系统默认的界面布局适用于建模等标准应用，同时系统还内置了多种界面布局，针对不同的工作环境需要，供用户快速地切换使用。例如有专门针对 Animation (动画)、Compositing (合成) 和 UV Editing (UV 编辑) 等应用的窗口布局模式。只需要在主标题栏上单击如图 2-3 左图所示的小按钮，即可在下拉菜单中选择这些对应工程的窗口布局模板，当然也可以使用快捷键  $\text{Ctrl} + \leftarrow$  或  $\text{Ctrl} + \rightarrow$  来对布局做快速切换。

在 Blender 中，每一个以“.Blend”为后缀的文件都是一个 Project（工程），每一个工程可以包含多个 Scene（场景），可以使用不同的窗口布局来编辑同一个独立的 Scene（场景）世界。一个工程中默认只有一个 Scene（场景），我们可以为同一个工程添加多个 Scene（场景），如图 2-3 中图所示。只需要单击 Scene（场景）管理中的十字星箭头，即可为当前工程添加一个新的 Scene（场景）。



图 2-3 Layout（布局）和 Scene（场景）管理

如果希望能对启动时使用的界面布局进行自定义，可以在完成布局设置后，单击 Ctrl + U 来保存你的设置，如图 2-3 右图所示。这样下次运行 Blender 时，系统会默认加载保存的自定义窗口布局。

### 2.1.3.3 自定义布局

Blender 的界面布局具有很高的自由度，每一个窗口的尺寸和位置均可任意地调节，可以根据需求将整个窗口划分成不同类型的布局结构。

首先将鼠标移动至两个窗口的分界处，待鼠标箭头变成平行的双向箭头时，即可按住 LMB 拖动窗口做水平移动，同理也可对窗口做上下移动操作，如图 2-4 左图所示。

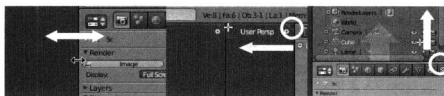


图 2-4 缩放、分离和合并窗口操作

如果需要分割窗口，可将鼠标移动到窗口右上方的箭头位置，待鼠标箭头变成十字星时即可向左或者向下拖动，从原窗口中分割出一个新的窗口，如图 2-4 中图所示。

合并窗口的操作和分割操作相反，待鼠标箭头变成十字星时即可向右或者向上拖动，如图 2-4 右图所示。被合并的两窗口之间会出现一个巨大的半透明箭头，箭头方向将指向被合并的窗口。在任何窗口操作过程中，都可以单击 ESC 取消操作。

标题栏可放置在窗口的顶部、底部，或者被隐藏起来。使用移动窗口的操作方式来向下移动标题栏，如图 2-5 左图所示。当标题栏缩小到一定比例时，便会被自动隐藏起来，如果想恢复显示，可单击窗口右下角的小十字星，如图 2-5 中图所示。在标题栏上单击鼠标右键，将弹出标题栏的操作菜单，如图 2-5 右图所示。选择 Flip to Top（反转至顶部），可将标题栏从窗口的底部移动至顶部。菜单中的 Maximize Area（区域最大化）功能，可将当前窗口最大化至全屏显示，也可以使用快捷键 Ctrl + ↑ 来完成窗口的最大化操作。

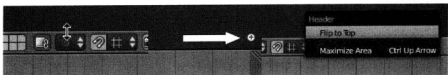


图 2-5 隐藏、恢复和移动标题栏操作

## 2.2 用户参数设置

在视图窗口中，单击组合键 **Ctrl + Alt + U** 可弹出 **User Preferences**（用户参数）设置窗口，在这里可以调整 **Interface**（界面）、**Editing**（编辑方式）、**Input**（快捷键）、**AddOns**（插件）、**Themes**（界面主题）、**Files**（文件）系统和 **System**（系统）等相关的参数设置。完成设置后，需要单击左下角的 **Save As Default**（存储为默认）按钮，才能保存所有的更改。如果需要重置系统的默认设置，可进入 Blender 的根目录 `blender\config` 文件夹，删除里面的“`startup.blend`”文件即可。

### 2.2.1 用户界面设置

如图 2-6 所示为 **Interface**（用户界面）设置面板。

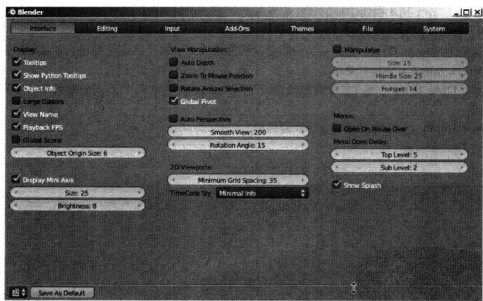


图 2-6 Interface（用户界面）设置面板

#### 2.2.1.1 Display（显示）设置

**Tooltips/Show Python Tooltips**：激活后，当把鼠标移至任意选项或按钮上时，会弹出一个功能帮助的信息提示框。

**Object Info**：开启后，在 3D 视图窗口的左下角，将显示当前物体的名称和时间、帧数等信息。

**Large Cursors**：启用大号的光标。

**View Name**：开启后，在 3D 视图窗口的左上角将显示视图名称等信息。

**Playback FPS**：在动画回放中开启帧数的显示。

**Global Scene**：全局场景选项，使当前场景中的物体可在其他场景中实现全局调用。

**Object Origin Size**：用于控制物体的原心尺寸，直径范围为 4 ~ 10 个 Blender 单位。

**Display Mini Axis:** 控制3D窗口左下角的小坐标尺寸, Size 用于调节大小, Brightness 则用于调节透明度。

### 2.2.1.2 View Manipulation (视图控制) 设置

**Auto Depth:** 开启视图的自动 Z 缓冲深度检测。

**Zoom To Mouse Position:** 开启后, 使用 MMB 对视角做旋转等操作时, 画面将以鼠标光标为焦点。

**Rotate Around Selection:** 开启后, 使用 MMB 对视角做旋转等操作时, 画面将以选择物体为焦点。

**Global Pivot:** 使各窗口中 Pivot 旋转坐标的显示模式同步。

**Auto Perspective:** 开启后, 视图将会在使用小键盘 Num1 ~ 9 切换时, 自动取消透视图。

**Smooth View:** 控制视角画面切换时, 动画过渡效果的时间长度。

**Rotation Angle:** 使用小键盘的 Num4、6、8、2 作为视角角度变换时的步进角度。

**2D Viewports:** 修改2D视图中的网格尺寸最小值, 例如 Timeline 时间线和 Dope Sheet 中的背景网格。

### 2.2.1.3 Manipulator (控制器) 设置

**Size:** 修改控制器的整体尺寸。

**Handle Size:** 调节控制杆的手柄尺寸。

**Hotspot:** 调节用于激活手柄的热区范围。

### 2.2.1.4 Menus (菜单) 设置

**Open On Mouse Over:** 激活后, 只需将鼠标滑动至菜单名上, 即可弹出当前菜单, 而无需使用 LMB 单击菜单名来打开菜单。

**Menu Open Delay:** 修改菜单弹出的延时时间。Top Level 和 Sub Level 分别调节一级菜单和二级菜单。

**Show Splash:** 是否在启动时显示欢迎界面。

## 2.2.2 编辑设置

如图 2-7 所示为 Editing (编辑) 属性设置面板。

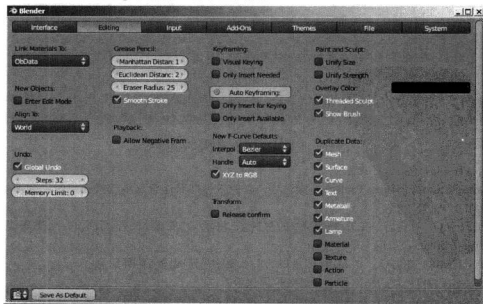


图 2-7 Editing (编辑) 属性设置面板



### 2.2.2.1 Link Materials To (材质链接方式) 设置

当我们在 Blender 中创建一个 Object (物体) 时, 它的材质等属性数据就会自动链接至 Scene (场景) 数据库中, 图 2-8 左图即为场景的树形数据结构图。

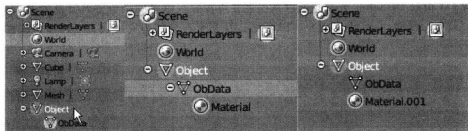


图 2-8 材质在数据结构中的不同链接方式

当选择链接方式为 ObData 时, 为一个网格物体添加一个 Material 材质属性, 其材质就会被创建为网格属性的一个子物体类, 作为网格的一个数据属性, 如图 2-8 中图所示, 表示了网格与材质的从属关系。

如果选择链接方式是 Object, 材质将作为一个并列的物体数据方式来创建, 如图 2-8 右图所示, 这时网格和材质将不再存在从属关系。

### 2.2.2.2 New Objects (新建物体) 设置

**Enter Edit Mode:** 激活此选项后, 当新建一个物体时, 系统会自动进入其编辑模式。

**Align To:** 设置物体在被创建时所默认使用的坐标系, 选项 World 将在世界坐标系上创建物体, 选项 View 则会将物体创建在与屏幕垂直的视图上。如图 2-9 所示, 左边的平面和大圣是在 World 模式下创建的效果, 而右边则为 View 模式下创建的效果。

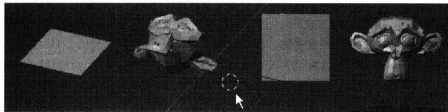


图 2-9 不同视角下创建物体的效果

### 2.2.2.3 Undo (撤销) 设置

**Global Undo:** 使用全局撤销模式, 步骤数据将会保存在内存中。

**Steps:** 设置存储的可撤销步骤数量。

**Memory Limit:** 存储撤销数据的内存设置, 0 为不限制。

### 2.2.2.4 Grease Pencil (视图绘制铅笔) 设置

**Manhattan Distance:** 设置绘制时笔触的绘制像素范围。

**Euclidean Distance:** 设置连续笔触间的间距。

**Eraser Radius:** 设置橡皮擦的尺寸。

**Smooth Stroke:** 设置平滑的路径参数。

### 2.2.2.5 Key framing (关键帧) 设置

**Visual Keying:** 设置是否为约束物体自动添加视图关键帧。

**Only Insert Needed:** 只插入当前修改的属性关键帧。

**Auto Keyframing:** 是否开启自动插入关键帧的功能, 当激活后可设置其附属功能 Only Insert for Keying Set 和 Only Insert Available, 分别控制插入帧的方式。

**New F-Curve Defaults:** F 曲线的默认控制, 可设置控制杆的调节方式分别为 Bezier/Liner/Constant 贝济埃/线性/非线性。如图 2-10 所示, 从左至右分别为贝济埃、线性和非线性曲线的形态。

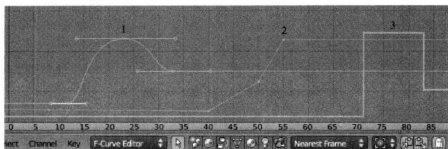


图 2-10 Bezier/Liner/Constant 三种曲线的形态

#### 2.2.2.6 Paint and Sculpt (绘制与雕刻) 设置

**Unify Size/Strength:** 是否使用统一的笔触尺寸和强度。

**Overlay Color:** 设置修改笔触时的背景色。

**Threaded Sculpt:** 使雕刻支持多线程的 CPU 计算。

**Show Brush:** 设置在雕刻的过程中显示笔刷。

#### 2.2.2.7 Duplicate Data (数据复制) 设置

用于定义默认的数据复制模式, 当复制物体时, 系统会根据当前激活的属性选项, 为物体的数据做选择性复制。如图 2-11 左图所示, 在激活 Mesh 选项后对物体做复制操作时, 可以看到物体和其网格属性都产生了全新的复制体。而当取消 Mesh 复制选项时, 如图 2-11 右图所示, 只有物体数据被复制了, 而网格属性数据并没有被复制, 只是被重复地调用至新的复制物体中, 作为子属性。这里使用的数据复制模式都是基于直接复制, 快捷键为 Shift + D, 而不是快捷键为 Alt + D 的关联复制, 后者将默认为新物体复用全部的属性数据。

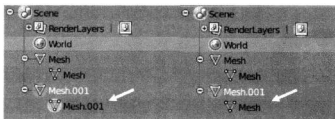


图 2-11 数据的复制模式

### 2.2.3 输入控制设置

如图 2-12 所示为 Input (输入控制) 的设置面板。

**Presets:** 键盘预设, 用于选择预设的快捷键设置, 可以使用其他软件的快捷键, 例如 Maya 等。



图 2-12 Input (输入控制) 设置面板

### 2.2.3.1 Mouse (鼠标) 设置

**Emulate 3 Button Mouse:** 如果使用的是双键鼠标或苹果鼠标, 可以开启这个选项来模拟三键鼠标的功能, 详细使用方法参考前面介绍的鼠标使用规则。

**Continuous Grab:** 使鼠标在控制拖动范围超出了画面时, 启用连续的滚动模式。例如拖动一个物体一直向右移动时, 当鼠标光标到达窗口最右端, 将会自动移动到窗口的最左端, 使得光标的移动动作可以无限地连续下去。

**Select with:** 决定使用鼠标左键还是右键来选择物体, 默认为右键。

**Double Click:** 设置用于判断鼠标双击动作的间隔时间。

**Emulate Numpad:** 如果键盘没有小键盘区, 开启这个功能后数字键 0~9 可用来代替 NumPAD。

**Orbit Style:** 习惯使用 Maya 的用户可开启 Turntable 来设置视图控制方式。

**Zoom Style:** 修改控制视图的缩放方式。

**Invert Zoom Direction/Wheel Zoom Direction:** 控制 Ctrl + RMB 和 MMB 对视图进行缩放的方式。

### 2.2.3.2 Key Configuration (自定义快捷键) 设置

Blender 允许用户自定义操作快捷键, 修改时只需要单击对应快捷键组合旁边的 Edit 按钮, 在下面的选项中更改鼠标和键盘的组合即可, 如图 2-13 所示。

**Restore:** 重置按钮可恢复修改的快捷键至默认状态。下面菜单中的每一栏都是一个快捷键组合, 单击左边的小三角可扩展这个菜单, 菜单中就是当前功能的具体快捷键选项了。

**Keyboard:** 设置快捷键为单独的按键还是组合的按键。

**Mouse:** 可选择 LMB、MMB 和 RMB 动作, 也可以与 Ctrl 和 Alt 等功能键组成组合键。

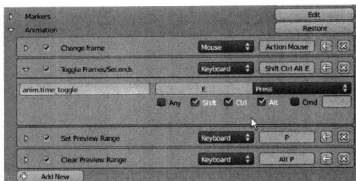


图 2-13 Key Configuration (自定义快捷键) 设置面板

**Tweak:** 设置拖拽动作，其中 Text Input 为键盘的输入控制。

完成修改之后需要单击 Save As Default 保存设置，否则下次启动 Blender 时将重置你的设置。也可单击 Export Key Configuration 导出并保存快捷键的设置文件，同样也可以使用 Import Key Configuration 功能来导入已设置好的快捷键设置文件。

## 2.2.4 附加脚本设置

Blender 可使用 Add-Ons 来完成自定义的脚本功能，如图 2-14 所示为 Add-Ons (附加脚本) 设置面板，具体的脚本使用将在第 11 章中重点讨论。



图 2-14 Add-Ons (附加脚本) 设置面板

## 2.2.5 主题设置

在主题设置面板中，可以自定义 Blender 的界面，包括菜单颜色、字体大小和滑块透明度等如图 2-15 所示。Blender 的 UI 修改是实时反应的，在修改设置的同时，就能立刻在主窗口中看到所做的改动了。

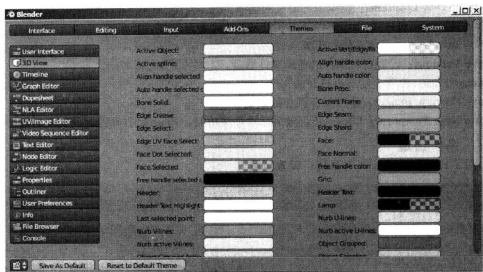


图 2-15 Themes（主题）设置面板

## 2.2.6 文件系统

在 File（文件）系统面板中，可以设置一些常用的文件夹路径，例如字体库、材质库、渲染输出、音频库和脚本库等目录，如图 2-16 所示。合理的路径设置对工程管理很重要，它能更好地帮助你合理地设计和组织项目文件，协调团队合作。

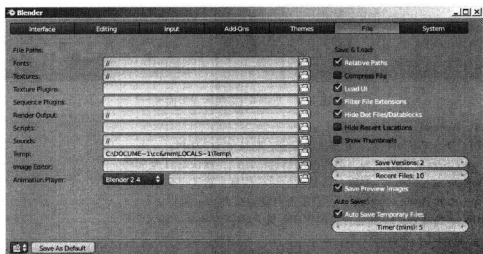


图 2-16 File（文件）系统设置面板

### 2.2.6.1 Save & Load（存储与载入）设置

**Relative Paths:** 激活后 Blender 会在读取文件时自动使用相对路径。

**Compress File:** 激活文件压缩功能。

**Load UI:** 打开用户工程时，自动载入存储的用户界面布局。

**Filter File Extensions:** 启用文件浏览器中的文件类型过滤器。

**Hide Dot Files/Datablocks:** 隐藏以（. \*）命名结构的文件和数据块。

**Hide Recent Locations:** 隐藏最近浏览的选项。

**Show Thumbnails:** 使用缩略图的方式打开图片或视频文件。

### 2.2.6.2 Auto Save (自动存储) 设置

**Save Versions:** 设置自动保存的备份文件数量。

**Recent Files:** 设置显示最近打开的文件个数。

**Save Preview Images:** 是否保存图片的预览。

**Auto Save Temporary Files:** 是否激活自动存储功能, Timer 是自动保存的间隔时间。

## 2.2.7 系统设置

如图 2-17 所示为 System (系统) 设置面板。

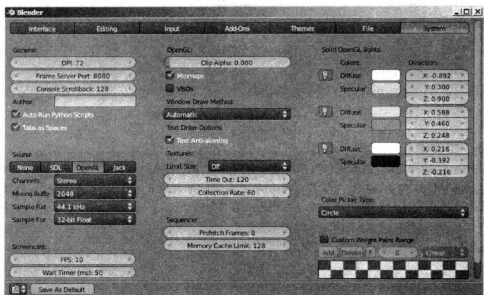


图 2-17 System (系统) 设置面板

### 2.2.7.1 General (常规) 设置

**DPI:** Dots Per Inch 显示分辨率的设置, 默认值为 72DPI。

**Frame Server Port:** 允许调用的网络通信端口, 默认端口为 8080。

**Console Scrollback:** 控制台可显示的运行信息, 默认可显示 128 行。

**Auto Run Python Scripts:** 设置是否自动运行 Python 脚本。

**Tabs as Spaces:** 设置是否自动将 Tab 输入转换为 Spaces 输入。

### 2.2.7.2 Sound (音频) 设置

**SDL:** Simple Direct Media Layer 简易直控媒体层驱动, 推荐应用于序列编辑。

**OpenAL:** Open Audio Library 跨平台音效 API 驱动, 推荐应用于游戏引擎开发。

**Jack:** Jack Audio Connection Kit 开源专业音效平台, 推荐应用于专业音频开发。

**Channels:** 输出音轨声道, 最高可调节至 7.1 环绕声道。

**Mixing Buffer:** 设置混音的缓存大小, 默认为 2048。

**Sample Rate:** 设置采样率为 44.1 kHz 到 192 kHz。

**Sample For:** 设置使用的采样格式, 例如 32 位浮点式。

### 2.2.7.3 OpenGL 设置

**Clip Alpha:** 设置纹理预览时边缘阈值的透明度。

**Mipmaps:** 开启后, 可以优化贴图显示, 但是会消耗一部分内存资源。

**VBOs:** 启用 Vertex Buffer Objects 顶点缓冲体, 用于优化视图渲染。

**Window Draw Method:** 界面显示模式选项。

**Test Anti-aliasing:** 开启文字的抗锯齿优化显示模式。

#### 2.2.7.4 Textures (贴图) 设置

**Limit Size:** 是否使用贴图尺寸限制。

**Time Out:** 贴图在更新后的刷新时间。

**Collection Rate:** 贴图回收的刷新时间。

#### 2.2.7.5 Sequencer (序列编辑器) 设置

**Screen cast:** 预览视频的设置, 调节窗口回放时的 FPS 帧数。

**Prefetch Frames:** 设置预览时预渲染的帧数。

**Memory Cache Limit:** 序列编辑器的缓存值设置。

#### 2.2.7.6 Solid OpenGL lights (场景实时灯光) 设置

用于设置 3D 视图中实时灯光的颜色等属性, 其中 Diffuse 用于设置散射光的方向和颜色, Specular 则用于设置反射光的方向和颜色。系统默认只开启了两盏灯, 也可以开启第三盏灯照明, 达到自下向上的补光效果。如图 2-18 所示, 左图为两盏灯下的模型预览, 右图为三盏灯下的预览效果。

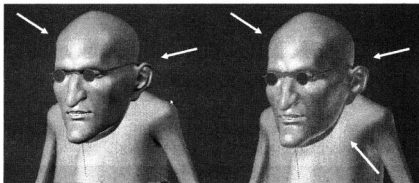


图 2-18 场景实时灯光效果

#### 2.2.7.7 Color Picker Type (拾色器) 设置

这里用于设置不同类型的拾色器面板属性, 如图 2-19 所示, 从左至右分别是 Circle、Square (SV + H)、Square (HS + V) 和 Square (HV + S) 类型的拾色器面板。

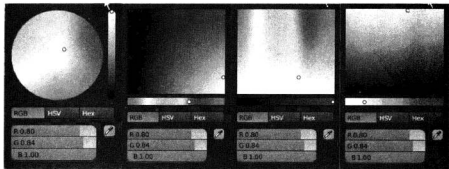


图 2-19 不同类型的拾色器面板

**Custom Weight Paint Range:** 自定义权重颜色的过渡色带。

## 2.3 窗口类型

Blender 的界面可以任意切割出多个编辑窗口，每个编辑窗口拥有 16 种类型，每种类型都提供了不同的界面和功能。窗口标题栏最左边的小图标显示了当前窗口的类型，单击图标即可弹出编辑窗口的类型菜单，如图 2-20 所示。使用 LMB 选择指定的窗口类型即可。

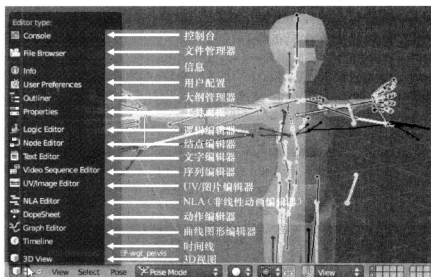


图 2-20 编辑器窗口

### 2.3.1 三维视图

3D View（三维视图）窗口是最常用的窗口之一，它用于显示当前所创建的 3D 场景，同时提供了大部分建模时使用的工具菜单和属性菜单，如图 2-21 所示。

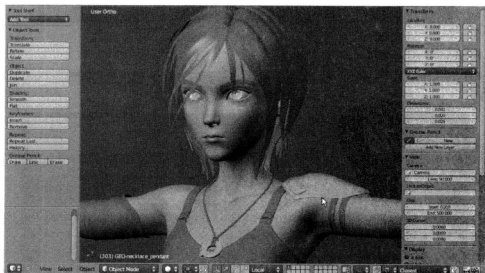


图 2-21 3D View（三维视图）编辑窗口



### 2.3.1.1 Tool (工具) 菜单

以 Object Mode (物体模式) 为例, 在窗口中单击快捷键 T 可以调出图 2-21 左侧的 Tool 工具面板, 这里面包括了一些常用的建模工具, 下面分别进行介绍。

**Transform:** 形变工具, 包括 Translate (变形)、Rotate (旋转)、Scale (缩放工具)。

**Object:** 物体工具, 包括 Duplicate (复制)、Delete (删除)、Join (合并工具)。

**Shading:** 着色预览工具, 包括 Smooth (平滑)、Flat (棱角工具)。

**Keyframes:** 关键帧工具, 包括 Insert (插入)、Remove (删除工具)。

**Repeat:** 历史工具, 包括 Repeat Last (重复最后动作)、History (历史动作工具)。

**Grease Pencil:** 画线工具, 包括 Draw (绘制)、Line (线条)、Erase (橡皮工具)。

### 2.3.1.2 Properties (属性) 菜单

在窗口中单击快捷键 N 可调出属性面板, 这里包括了物体和视图的属性设置, 如图 2-22 所示。

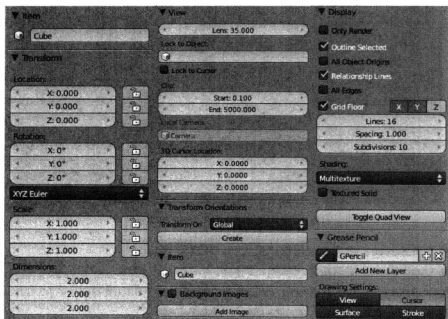


图 2-22 视图属性面板

**Item:** 物体名称, 可在文本框中对物体重命名。

**Transform:** 物体属性, 包括 Location (位置)、Rotation (转动)、Scale (比例)、Dimensions (尺寸)。

**View:** 视图属性, 包括 Lens (视图焦距)、Lock to Object (视图对物体锁定)、Lock to Cursor (视图对光标锁定)、Clip (动画区域属性)、Local Camera (默认使用的摄像机名称)、3D Cursor (指针属性)。

**Transform Orientations:** 坐标变换属性, 包括 Transform Ori 坐标系切换等属性。

**Background Images:** 背景图属性, 包括 Add Image (添加新的背景图) 等属性。

**Display:** 显示属性, 包括 Only Render (仅显示渲染物体)、Outline Selected (显示在大纲编辑器中选择的物体)、All Object Origins (显示所有物体的原心)、Relationship Lines (显示物体间的关系)、All Edges (显示所有的线框边)、Grid Floor (网格显示属性)、Shading (着色预览模式)、Toggle Quad View (快速划分视图)。

**Grease Pencil:** 视图绘制属性, 包括 Add New Layer (新建绘图图层)、Opacity (画笔不透明

度)、Thickness (画笔粗细)、Onion Skinning (洋葱皮模式)、Frames (可见帧数)、Delete Frame (清除)、Convert (转换成网格物体) 等选项。

### 2.3.1.3 View (视图) 菜单

如图 2-23 所示为 3D 视图窗口的 View (视图) 菜单。

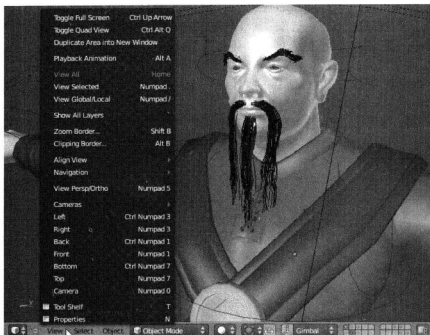


图 2-23 3D 视图窗口的标题栏菜单

**Toggle Full Screen:** 将当前视图窗口最大化, 快捷键为 Ctrl + ↑。

**Toggle Quad View:** 快速切分三视图, 快捷键为 Ctrl + Alt + Q。

**Duplicate Area into New Window:** 为当前窗口复制出一个新的窗口。

**Playback Animation:** 动画回放, 快捷键为 Alt + A。

**View All:** 将视图缩放至全部物体可见, 快捷键为 Home。

**View Selected:** 将视图转向所选物体, 快捷键为 Num .。

**View Global/Local:** 切换全局和本地查看, 快捷键为 Num /。

**Show All Layers:** 查看全部图层, 快捷键为 ~。

**Zoom Border:** 视图缩放至框选区域, 快捷键为 Shift + B。

**Clipping Border:** 视图只显示框选区域, 快捷键为 Alt + B。

**Align View:** 视图排列方式, 其中 Align View to Selected 为视图视角的对齐方式, 如图 2-24 所示为视图转换方式的二级菜单, 其中列举了各视图间相互转换时的快捷键。

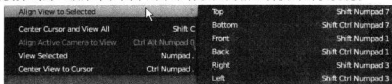


图 2-24 Align View (视图排列) 的二级菜单

**Center Cursor and View All:** 将光标重置到原点位置, 快捷键为 Shift + C。

**Center View to Cursor:** 将当前视图切换至以光标为中心的缩放画面。

**Navigation:** 导航模式菜单, 如图 2-25 所示为导航模式的二级菜单。其中, 可以对视图做一定角度的视角旋转, 快捷键分别为 Num2、4、6、8。

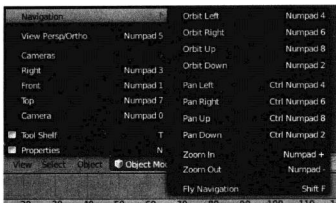


图 2-25 Navigation (导航模式) 的二级菜单

**Fly Navigation:** 飞行预览模式, 将镜头视图窗口切换至可飞行的预览状态, 快捷键为 Shift + F。

**View Persp/Ortho:** 透视图与正交视图间的切换, 快捷键为 Num5。如图 2-26 所示, 左图为透视图的效果, 右图为同一个视角下的正交视图效果。可以从网格的形状和物体的边缘角度, 区分两种视图的差别。

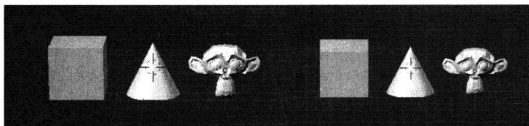


图 2-26 透视与正交视图比较

**Cameras:** 激活所选摄像机为默认的镜头视图, 快捷键为 Ctrl + Num0。

**Right:** 切换当前视图至侧视图视角, 快捷键为 Num3。

**Front:** 切换当前视图至前视图视角, 快捷键为 Num1。

**Top:** 切换当前视图至顶视图视角, 快捷键为 Num7。

**Camera:** 切换当前视图至摄像机视图视角, 快捷键为 Num0。

**Tool Shelf:** 调出工具菜单, 快捷键为 T。

**Properties:** 调出属性菜单, 快捷键为 N。

#### 2.3.1.4 Select (选择) 菜单

如图 2-27 所示为 3D 视图窗口的 Select (选择) 菜单。

**Select Patten:** 使用输入查找来选取物体。如图 2-28 所示为输入物体名称 Camera 来查找当前场景中摄像头的结果。

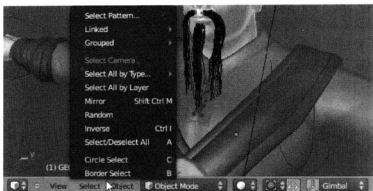


图 2-27 Select (选择) 菜单



图 2-28 Select Patten (查找选择) 效果

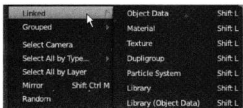


图 2-29 Linked (相似选择) 的二级菜单

**Linked:** 相似选择, 快速选择拥有相同的材质、贴图或者粒子组等属性的物体, 快捷键为 Shift + L。如图 2-29 所示为相似选择方式的二级菜单。

**Grouped:** 群组选择, 快速选择具有通用群组关系的物体。如图 2-30 所示为群组选择功能的二级菜单, 可以使用这个功能快速选择不同类型的子物体、父物体和同类型物体等, 快捷键为 Shift + G。

**Select Camera:** 快速选择当前默认的摄像机物体。

**Select All by Type:** 选择相同类型的物体。如图 2-31 所示, 可以快速地选择同类属性的网格、曲线或者骨骼等物体。

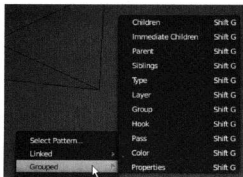


图 2-30 Grouped (群组选择) 的二级菜单

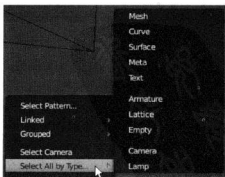


图 2-31 Select All by Type (相同类型选择) 的二级菜单

**Select All by Layer:** 选择可视层中的所有可见物体。

**Mirror:** 选择当前物体的镜像命名物体, 例如当前物体名为 Cube.L, 那么单击快捷键 Shift + Ctrl + M, 系统将自动选择场景中存在的物体 Cube.R。

**Random:** 随机选择。

**Inverse:** 反向选择, 快速选择当前未选中的其他物体, 快捷键为 Ctrl + I。

**Select/Deselect All:** 全选, 快捷键为 A。

**Circle Select:** 笔触选择模式, 快捷键为 C。激活后鼠标将变成一个圈, 使用 LMB 做选择操作, MMB 调节选择笔触的尺寸。

**Border Select:** 框选模式, 快捷键为 B。区别于笔触选取, 该模式需要使用鼠标拖拽出一个矩形范围框。如图 2-32 所示为两种选择方式的区别, 左图为 Circle 选择模式, 右图为 Border 选择模式。

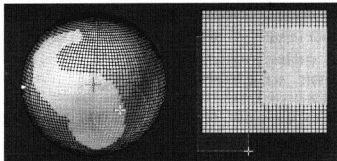


图 2-32 Circle (笔触) 选择和 Border (框选) 效果

### 2.3.1.5 Object (物体) 菜单

如图 2-33 所示为 3D 视图窗口的 Object (物体) 菜单。

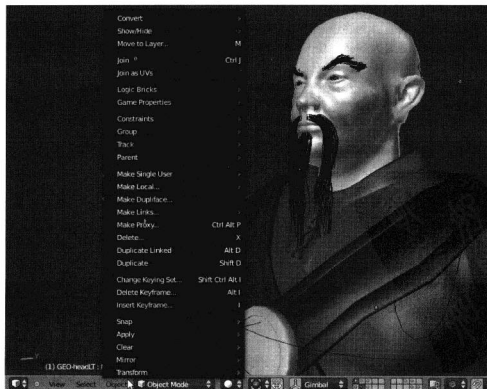


图 2-33 Object (物体) 菜单

**Convert:** 类型转换, 将曲线转换为网格物体。或者将网格转换成曲线物体。如图 2-34 所示为其二级菜单, 快捷键为 Alt + C。

**Show/Hide:** 隐藏被选择物体在视图中的显示, 快捷键为 H。如果要恢复被隐藏物体的显示, 可使用快捷键 Alt + H。如果要将被选择的物体隐藏起来, 可使用快捷键 Shift + H。如图 2-35 所示为其二级菜单。



图 2-34 Convert (类型转换) 的二级菜单



图 2-35 Show/Hide (显示/隐藏) 的二级菜单

**Move to Layer:** 将物体移动到指定的图层, 快捷键为 M。

**Join:** 合并多个物体, 快捷键为 Ctrl + J。

**Join as UVs:** 复制目标物体的 UV。

**Constraints:** 弹出约束创建菜单, 快捷键为 Shift + Ctrl + C。如果要清除约束, 快捷键为 Ctrl + Alt + C。如图 2-36 所示为其二级菜单。

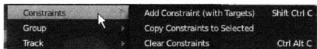


图 2-36 Constraints (约束创建) 的二级菜单

**Group:** 群组工具, 如图 2-37 所示为其二级菜单。如果要多个物体加入至新的群组, 快捷键为 Ctrl + G。将选择的物体从群组中移除, 快捷键为 Ctrl + Alt + G。将选择的物体添加到目标组当中, 快捷键为 Shift + Ctrl + G。将选择的物体从当前组中移除, 快捷键为 Shift + Alt + G。



图 2-37 Group (群组) 的二级菜单

**Track:** 建立跟随关系, 如图 2-38 所示为其二级菜单, 快捷键为 Ctrl + T, 取消跟随关系快捷键为 Alt + T。

**Parent:** 建立父子关系, 如图 2-39 所示为其二级菜单, 快捷键为 Ctrl + P, 取消父子关系的快捷键为 Alt + P。



图 2-38 Track (跟随) 的二级菜单



图 2-39 Parent (父子) 的二级菜单

**Make Single User:** 取消属性的复用。如图 2-40 的上图所示, 两个物体分别被赋予了同样的材质, 当选择物体 Mesh.001 并单击 Make Single User 后, 材质将被取消复用, 而被复制建立出一个新的材质, 并被添加一个新的后缀名作为重命名, 如图 2-40 的下图所示。

**Make Local:** 将物体本地化, 常用于将从其他工程引用的物体独立化至本地工程, 快捷键为 L。

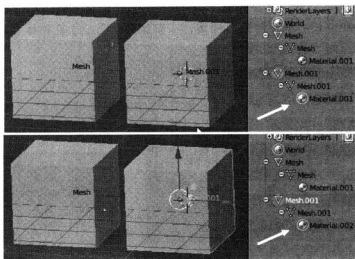


图 2-40 取消属性的复用

**Make Dupliface:** 为当前物体建立一个镜像体。

**Make Links:** 建立链接关系，为物体在不同场景之间建立属性的链接关系，如图 2-41 所示为其二级菜单。

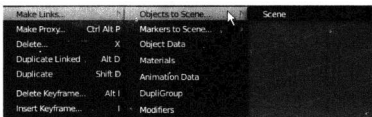


图 2-41 Make Links（建立链接）的二级菜单

**Make Proxy:** 为当前链接的库物体建立一个本地空白数据对象。

**Delete:** 删除，快捷键为 X。

**Duplicate Linked:** 建立关联复制，快捷键为 Alt + D。在对 Mesh 使用关联复制后，物体的所有属性并没有被复制而是被链接至新的物体。如图 2-42 所示，可以从大纲管理器中看到，复制出的新物体 Mesh.001 依然使用了和原物体 Mesh 属性相同的子网格属性 Mesh.004，这样当修改 Mesh 时，Mesh.001 就会在相应属性上做出同步的更新修改。

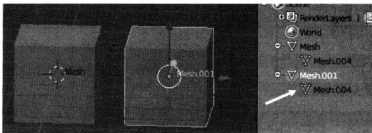


图 2-42 Duplicate Linked（关联复制）效果

**Duplicate:** 复制新物体, 快捷键为 Shift + D。与关联复制不同, 在对 Mesh 做复制操作后, 新的 Mesh.001 并没有继承原物体的属性, 而是派生出一个新物体和属性, 如图 2-43 所示。

**Delete Keyframe:** 清除关键帧, 快捷键为 Alt + I。

**Insert Keyframe:** 插入关键帧, 快捷键为 I。

**Snap:** 吸附功能, 如图 2-44 所示为其二级菜单, 快捷键为 Shift + S, 具体使用会在后面的 Edit Mode (编辑模式) 中讨论。

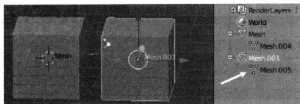


图 2-43 Duplicate (复制) 效果

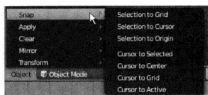


图 2-44 Snap (吸附) 的二级菜单

**Apply:** 应用属性的修改, 快捷键为 Ctrl + A。如图 2-45 左图所示, 新建一个 Scale (缩放比例) 默认为 1 的 Mesh 物体, 接下来将其缩放为原来的一半大小, 这时 Scale 变成了 0.5, 如图 2-45 中图所示。当单击 Apply Scale 应用属性之后, Scale 属性将自动应用缩放的修改, 把属性值重置为 1, 如图 2-45 右图所示。

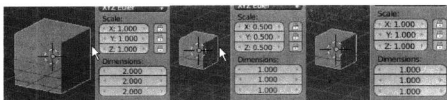


图 2-45 Apply (应用) 属性

**Clear:** 它是与 Apply (应用) 相反的一个功能, 用于清除对物体做的形变等操作。其中位移重置快捷键为 Alt + G, 转动重置为 Alt + R, 缩放重置为 Alt + S, 原点重置为 Alt + O。如图 2-46 所示为 Clear (清除) 功能的二级菜单。

**Mirror:** 镜像工具, 将当前物体按坐标轴做镜像变化, 快捷键为 Ctrl + M, 如图 2-47 所示为其二级菜单。



图 2-46 Clear (清除) 的二级菜单



图 2-47 Mirror (镜像) 的二级菜单

**Transform:** 形变功能, 如图 2-48 所示为其二级菜单, 具体功能将在编辑模式中讨论。

### 2.3.1.6 Mode (模式) 选择

如图 2-49 所示, 我们可以在 3D 视图中为物体选择不同的编辑模式。单击模式选择旁边的箭头即可弹出模式菜单, 或者使用快捷键 Tab 做快速的切换。

**Object Mode:** 物体模式, 用于对场景中 Object (物体) 执行编辑操作, 例如移动和缩放等操作。



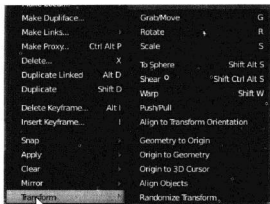


图 2-48 Transform (形变) 的二级菜单

**Edit Mode:** 编辑模式, 用于编辑物体的本身, 例如网格中的点线面调节等操作。

**Sculpt Mode:** 雕刻模式, 用于对模型做雕刻操作。

**Vertex Paint:** 顶点喷绘, 用于修改点与光影的交互影响效果。

**Texture Paint:** 贴图绘制模式, 用于在 3D 视图中实时绘制物体的贴图。

**Weight Paint:** 权重绘制模式, 用于为物体绘制权重。

#### 2.3.1.7 Viewport Shading (光影预览)

物体模式旁边的图标是 Viewport Shading (光影预览) 方式, 如图 2-50 所示。



图 2-49 物体模式类型



图 2-50 光影预览模式菜单

**Bounding Box:** 开启范围框显示模式。

**Wireframe:** 开启线框显示模式, 快捷键为 Z。

**Solid:** 开启 OpenGL 显示模式, 快捷键为 Shift + Z。

**Textured:** 开启贴图显示模式, 快捷键为 Alt + Z。

如图 2-51 所示, 从左至右分别是同一个物体在不同光影下的显示效果。



图 2-51 不同的光影视图预览

### 2.3.1.8 Pivot (控制杆) 选择

光影预览的旁边是控制杆选择菜单,如图 2-52 所示为不同控制杆的属性菜单。

控制杆的模式选择在建模和动画制作时很重要,可以从菜单选项中选择不同的控制方式,左侧的小图标形象地说明了不同选项下,控制杆和物体之间的位置关系。

**Bounding Box Center:** 将控制杆原心置于物体选择体的边框中心处。

**3D Cursor:** 将控制杆原心置于光标的位置。

**Individual Origins:** 将控制杆原心置于物体各自的原心处。

**Median Point:** 将控制杆原心置于多个物体的中央位置。

**Active Element:** 将控制杆原心置于最后一个激活的物体原心处。

如图 2-53 所示,从左至右分别表示了不同模式下,控制杆原心在被控物体上的位置效果。



图 2-52 Pivot (控制杆) 菜单

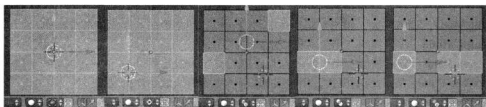



图 2-53 不同模式下控制杆的位置

Blender 有三种不同类型的控制杆,单击标题栏上的  按钮,可开启控制杆在场景中的显示。三种控制杆分别代表移动、旋转和缩放功能,如图 2-54 所示,从左至右分别为三种类型的控制杆在物体上的显示效果。

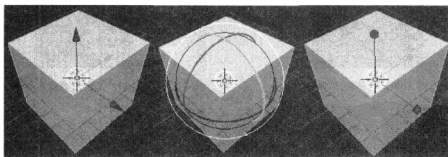


图 2-54 不同类型的控制杆

如图 2-55 所示为控制杆的显示模式菜单,菜单左边的 4 个小图标分别表示控制器的显示属性。右边的坐标菜单则提供了不同的控制器使用的坐标轴,包括 Global (全局) 坐标、Local (本地) 坐标、Gimbal (万向) 坐标、Normal (法向) 坐标和 View (视图) 坐标。

### 2.3.1.9 Layer (层) 选项

Layer 层工具是最常用的物体管理工具,它的工作原理和使用同 Photoshop 等软件的图层概念类似。可以将物体存放在不同的层上,用取消和选择的方式来隐藏和显示对应层上的物体,十分方便区分和管理较多物体的场景。如图 2-56 所示,Blender 提供了 20 个层,

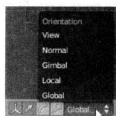



图 2-55 控制杆显示模式菜单

分别由 20 个小方块来表示。

Layer 管理器中的每个方块都代表一个层。当一个层被激活时这个方块会显示为灰色，未被激活的层将显示为白色。方块中的圆点表示该层至少存放了一个物体，而使用橙色标识出来的圆点，表示当前被选择激活的物体所在层。在为物体指定层时，可以首先选择物体，然后使用快捷键 M 将其移动到目标指定层上。当层菜单旁边的小按钮  被激活时，系统将自动使用当前场景层中的摄像机视角。

### 2.3.1.10 特殊编辑模式

Proportional Editing（比例衰减编辑）工具可以辅助执行一些特殊的选择操作，当点击 Enable 激活选项后，如图 2-57 所示，旁边会出现一个衰减方式列表菜单。这里列举了 Random（随机）、Constant（连续）、Linear（线性）、Sharp（尖锐）、Root（根部）、Sphere（球面）和 Smooth（平滑）共 7 种衰减模式。在后面的章节中将详细讨论每一种选择模式下的使用方式。



图 2-56 Layer（层）管理器

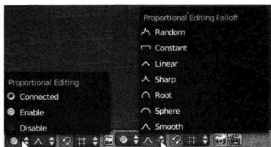






图 2-57 Proportional Editing（比例衰减编辑）和衰减方式

Snap（吸附）工具常在物体移动或对物体做精确变形时使用，如图 2-58 左图所示，在单击小磁铁图标  后可激活吸附功能。吸附模式分为 5 种，分别是 Volume（体积）、Face（面）、Edge（边）、Vertex（点）和 Increment（网格）。在选择一种吸附方式后，还需要在旁边的菜单中选择对应的吸附目标，如图 2-58 右图所示，包括 Active（激活的）、Median（正中的）、Center（中央的）和 Closest（最近的）。紧接着在旁边有两个图标： 表示是否应用目标物体的旋转角度， 则表示将吸附映射到目标的表面上。

标题栏上最右边的两个图标 ，分别表示 OpenGL 模式下的图片和动画渲染功能。功能被激活时，系统将使用场景灯光，实时地渲染当前窗口中的物体效果，如图 2-59 所示为 OpenGL 渲染效果。

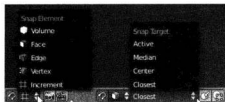


图 2-58 Snap（吸附）工具

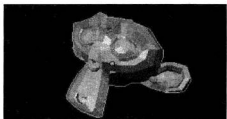


图 2-59 OpenGL 渲染效果

## 2.3.2 时间轴窗口

Timeline（时间轴）窗口，用于显示制作动画时使用的时间轴，如图 2-60 所示为 Timeline（时间轴）窗口的界面。



图 2-60 Timeline (时间轴) 窗口

可以滑动 MMB 修改时间轴上的显示范围,同时利用快捷键 **Ctrl + MMB** 对显示范围做缩放操作。时间轴上的绿线标识出了当前帧的时间位置,黄线标识了这一帧上建立有关键帧,单击 LMB 可以在轴上对时间做定位,将当前时刻指定至新的时间点位置。如图 2-60 所示的数字 35 标识了当前所在帧的位置,也可以手动输入以便在时间轴上快速定位。

时间轴上的数字默认表示了帧数,Start (起始) 选项可用于设置起始播放的帧位置,End (结束) 选项则设置最后一帧的时间轴位置,使用快捷键 **Home** 可以在视图中显示全部帧范围。如图 2-60 所示,表示当前的播放范围是从第 1 帧至第 250 帧。超出了设置范围的时间帧,背景将会使用深灰色来显示。标题栏上的播放按钮 用于控制动画回放。

激活红色圆圈按钮 可开启自动添加关键帧的功能,在对物体做移动或其他动画操作时,系统会自动在当前帧上添加相对应的关键帧,其中 选项可用于设置添加的关键帧类型。

### 2.3.2.1 View (视图) 菜单

如图 2-61 所示为 Timeline (时间轴) 窗口中的 View (视图) 菜单。

**Bind Camera to Markers:** 将当前激活的摄像机与 Marker 帧标识符绑定起来,可用于制作多镜头之间的自动切换,快捷键为 **Ctrl + B**。

**Only Selected channels:** 仅显示指定通道的时间轴属性。

**Show Frame Number Indicator:** 在时间轴上显示帧的数字。

**View All:** 自适应缩放显示从 Start 到 End 帧数的全部时间轴。

**Toggle Frames/Seconds:** 在时间显示方式秒和帧之间切换。

### 2.3.2.2 Frame (帧) 菜单

如图 2-62 所示为 Timeline (时间轴) 窗口中的 Frame (帧) 菜单。



图 2-61 View (视图) 菜单



图 2-62 Frame (帧) 菜单

**Auto-Keyframing Mode:** 自动添加/替换帧功能,作用与面板上的红色圆点按钮 相同。

**Set End Frame:** 设置动画范围的 End (结束) 帧,快捷键为 **E**。

**Set Start Frame:** 设置动画范围的 Start (起始) 帧,快捷键为 **S**。

**Delete Marker:** 删除帧标识符,快捷键为 **X**。

**Grab/Move Marker:** 选取和移动帧标识符,快捷键为 **G**。

**Duplicate Marker:** 复制帧标识符,快捷键为 **Shift + D**。

**Add Marker:** 添加帧标识符,快捷键为 **M**。

### 2.3.2.3 Playback (回放) 菜单

Playback (回放) 菜单定义了了在动画回放中, 参与动画效果的属性编辑器, 如图 2-63 所示为 Playback (回放) 属性菜单, 激活对应的编辑器, 在其中修改的动画属性将显示在场景回放中。



图 2-63 Playback (回放) 菜单

### 2.3.3 曲线图形编辑器

2.5 版本后的 Blender 采用了 F-Curve 曲线概念来管理物体的动画属性, 这使得物体对象的网格、材质和灯光等所有可调节的参数, 都可以通过控制其对应的 F-Curve 曲线来制作成动画效果。

F-Curve 是一个物体对象属性的最底层结构, 也是动画物体之间可通用的基础属性结构。如图 2-64 所示, 每一个 F-Curve (曲线) 都控制了一个物体的属性参数, 例如物体在 X 轴上的位移, 或者是在 Y 轴方向上的缩放比例等。一个物体的基本动画 Action (动作), 就是由多个 F-Curve (曲线) 封装组成, 最终实现一个完整的动画。同时, 不同的 Action 之间也可共享使用同一个 F-Curve 属性数据, 称做属性的复用。

Action 01

F-Curve/Data Path 路径名 (例如, location.X)

F-Curve/Data Path 路径名 (例如, location.Y)

F-Curve/Data Path 路径名 (例如, location.Z)

图 2-64 Action 与 F-Curve 的组成关系

Graph Editor (图形编辑器), 就是用于控制和修改物体 F-Curve 曲线属性的编辑环境。如图 2-65 所示, 在图形编辑器中, 每种 F-Curve (曲线) 都被记录在一个独立的通道中, 每个通道分别控制一个物体属性, 例如 X Scale 表示了物体在 X 方向上的缩放值, Y Location 为物体在 Y 方向上的位移等。

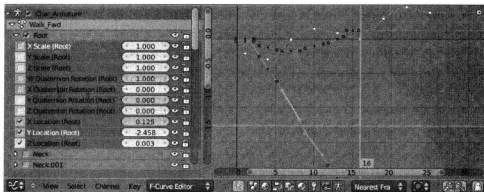


图 2-65 Graph Editor (图形编辑器) 编辑窗口

由于通道较多,为了在编辑的过程中易于区分,每一个通道的曲线都设置为不同的颜色,也可以在窗口中单击快捷键 N,对曲线颜色进行自定义修改。

窗口中的横坐标为时间轴,纵坐标为属性参数值,当改变一个 F-Curve 曲线在某一个时间点上的曲线形状时,属性参数值就会被设置为该曲线点在纵坐标上的数值。曲线编辑器的具体使用将在第 8 章深入讨论。

### 2.3.3.1 View (视图) 菜单

如图 2-66 所示为 Graph Editor (图形编辑器) 窗口中的 View (视图) 菜单。

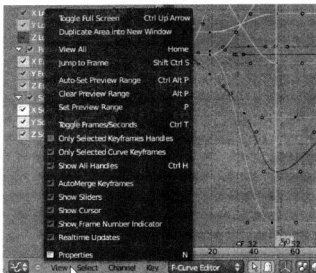


图 2-66 View (视图) 菜单

**Toogle Full Screen:** 将窗口最大化, 快捷键为  $\text{Ctrl} + \uparrow$ 。

**Duplicate Area into New Window:** 为当前窗口复制出一个新的窗口。

**View All:** 自适应缩放至可见全部有效范围的帧, 快捷键为 Home。

**Jump to Frame:** 快速定位至某一帧, 快捷键为  $\text{Shift} + \text{Ctrl} + \text{S}$ 。

**Auto-Set Preview Range:** 自动设置预览帧的范围, 帧范围由头尾两个关键帧的位置决定, 快捷键为  $\text{Ctrl} + \text{Alt} + \text{P}$ 。

**Clear Preview Range:** 清除预览帧范围, 快捷键为  $\text{Alt} + \text{P}$ 。

**Set Preview Range:** 自定义设置预览帧范围, 快捷键为 P。

**Toogle Frames/Seconds:** 时间轴在秒和帧之间做显示切换, 快捷键为  $\text{Ctrl} + \text{T}$ 。

**Only Selected Keyframes Handles:** 只显示被选择曲线点上的控制杆。

**Only Selected Curve Keyframes:** 只显示被选择的曲线。

**Show All Handles:** 显示全部控制杆, 快捷键为  $\text{Ctrl} + \text{H}$ 。

**AutoMerge Keyframes:** 自动合并最近的两个关键帧, 适用于对曲线做自动清理。

**Show Sliders:** 在通道名的旁边显示属性框。

**Show Cursor:** 开启在曲线上显示十字形光标。

**Show Frame Number Indicator:** 在时间轴上显示当前帧数。

**Realtime Updates:** 启用曲线与场景的实时更新显示。

**Properties:** 弹出属性框选项, 快捷键为 N。

### 2.3.3.2 Select (选择) 菜单

如图 2-67 所示为 Graph Editor (图形编辑器) 窗口中的 Select (选择) 菜单。

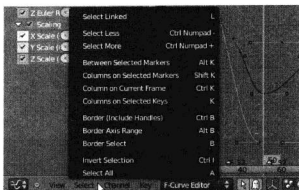


图 2-67 Select (选择) 菜单

**Select Linked:** 选择相似, 快捷键为 L。

**Select Less:** 递减步进选择, 快捷键为 Ctrl + Num -。

**Select More:** 递增步进选择, 快捷键为 Ctrl + Num +。

**Between Selected Markers:** 选择帧标识符之间的所有控制点, 快捷键为 Alt + K。

**Columns on Selected Markers:** 选择当前激活帧标识符上的所有控制点, 快捷键为 Shift + K。

**Column on Current Frame:** 选择当前激活帧上的所有控制点, 快捷键为 Ctrl + K。

**Columns on Selected Keys:** 选择与当前关键帧在同一时刻的所有帧, 快捷键为 K。

**Border Axis Range:** 框选当前轴范围上的所有关键帧, 快捷键为 ALT + B。

**Border Select:** 关键帧框选, 此时无法选中控制杆, 快捷键为 B。

**Border (Include Handles):** 关键帧带控制杆框选模式, 快捷键为 Ctrl + B。

**Invert Selection:** 反向选择, 快捷键为 Ctrl + I。

**Select All:** 全选, 快捷键为 A。

### 2.3.3.3 Channel (通道) 菜单

如图 2-68 所示为 Graph Editor (图形编辑器) 窗口中的 Channel (通道) 菜单。

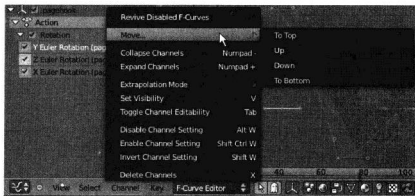


图 2-68 Channel (通道) 菜单

**Collapse Channels:** 收起所有通道, 快捷键为 Num -。

**Expand Channels:** 展开所有通道, 快捷键为 Num +。

**Extrapolation Mode:** 曲线延长模式, 快捷键为 Shift + E, 可选择 Constant (常量型) 和 Linear (线型) 模式。

**Set Visibility:** 隐藏除当前选择的其他所有通道, 快捷键为 V。

**Toggle Channel Editability:** 进入通道编辑模式, 快捷键为 Tab。

**Disable/Enable Channel Setting:** 关闭或开启通道选项, 例如通道的可视化选项和编辑保护模式, 快捷键为 Alt + W 和 Shift + Alt + W。

**Invert Channel Setting:** 反转当前的通道设置, 快捷键为 Shift + W。

#### 2.3.3.4 Key (帧) 菜单

如图 2-69 所示为 Graph Editor (图形编辑器) 窗口中的 Key (帧) 菜单。

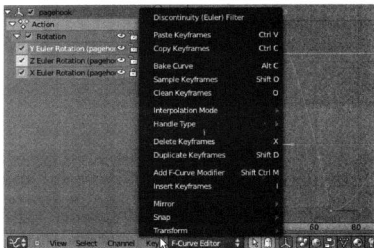


图 2-69 Key (帧) 菜单

**Paste Keyframes:** 粘贴帧, 快捷键为 Ctrl + V。

**Copy Keyframes:** 复制当前帧, 快捷键为 Ctrl + C。

**Bake Curve:** 曲线烘焙操作, 快捷键为 Alt + C, 烘焙后的曲线将删除所有的关键帧节点, 但是保持烘焙前由节点调节出的曲率和形状, 常用于精简动作的关键帧数据量。如图 2-70 左图所示为未烘焙的节点曲线; 烘焙后, 曲线将无法在关键帧位置做重编辑, 如图 2-70 右图所示。

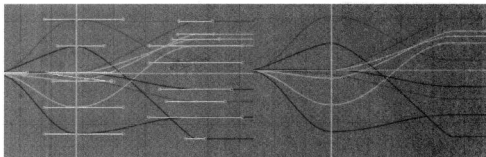


图 2-70 曲线 Baking (烘焙)

**Sample Keyframes:** 在两个关键帧区间, 自动为每帧的位置上添加一个关键帧, 快捷键为 Shift + O。



**Clean Keyframes:** 清除关键帧, 快捷键为 O。

**Interpolation Mode:** 修改曲线的插值类型, 快捷键为 Shift + T。

**Handle Type:** 修改控制杆的类型, 快捷键为 H。

**Delete Keyframes:** 删除关键帧, 快捷键为 X。

**Duplicate Keyframes:** 复制关键帧, 快捷键为 Shift + D。

**Add F-Curve Modifier:** 为曲线添加修改器, 快捷键为 Shift + Ctrl + M, 可选择的修改器包括, Generator (发生器)、Built-in Function (运算公式)、Envelope (封套形)、Cycles (循环型)、Noise (噪音)、Python (脚本控制)、Limits (极限) 控制和 Stepped (阶梯形)。借助修改器, 我们可以制作出一些特殊形状的曲线效果, 如图 2-71 所示。

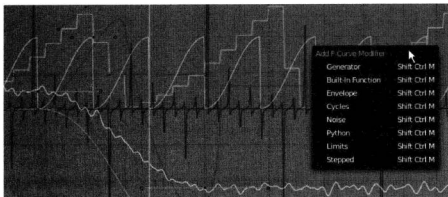


图 2-71 F-Curve Modifier (曲线修改器) 效果

**Mirror:** 制作曲线镜像, 快捷键为 Shift + M。

**Snap:** 启用曲线的吸附编辑模式, 快捷键为 Shift + S。

**Transform:** 曲线的常用形变编辑菜单, 包括, Grab 移动, 快捷键为 G; Extend 延伸, 快捷键为 E; Rotate 旋转, 快捷键为 R; Scale 缩放, 快捷键为 S。

### 2.3.3.5 模式菜单

在 Graph Editor 图形编辑器窗口的标题栏上, 可以看到一个模式菜单, 如图 2-72 所示。



图 2-72 模式菜单

模式菜单用于选择在 Graph Editor (图形编辑器) 窗口中显示的属性类型, 例如当激活 Mesh (网格) 属性, 即可在编辑器中对于 Mesh 相关的属性做 F-Curve (曲线) 化操作。

### 2.3.4 动作编辑器

调节 F-Curve (曲线) 是对属性的最底层修改, 当我们制作物体运动或骨骼动画时, 大量的曲线数据会让人眼花缭乱, 也无法通过调节 F-Curve (曲线) 来修改动作。这时可以直接进入 Dope Sheet (动作编辑器), 将修改曲线的方式换做直接调节整个 Action 元素, 将注意力只关注在当前调用的 Action (动作) 效果上。

Dope Sheet (动作编辑器) 可同时编辑多个 Action (动作) 关键帧, Summary (概述) 功能提供了一个大纲显示方式, 有助于观察多个 Action 叠加后的动作分布效果。Dope Sheet (动作编辑器) 的界面和菜单与曲线图形编辑器界面十分相似, 唯一不同的只有时间轴界面。如图 2-73

所示，窗口中横坐标为时间轴，纵坐标上的每一个通道代表一个物体的 Action（动作）属性。当这一秒上被添加一个关键帧时，对应的属性通道时间轴中会新建一个小菱形，这个菱形就是 Action（动作）块。移动和复制 Action（动作）块不能修改内部各曲线的属性，但是可以宏观地观察当前帧上 Action（动作）块的所有曲线效果。

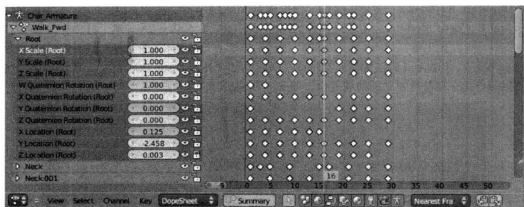


图 2-73 Dope Sheet（动作编辑器）编辑窗口

### 2.3.5 非线性动画编辑器

NLA（Non-Linear Animation，非线性动画编辑器）使用了非线性编辑原理，灵活地控制所有完成的 Action（动作）体，用于制作连续的动画，如图 2-74 所示为 NLA 的操作界面。

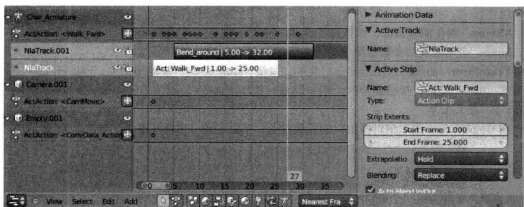


图 2-74 NLA（非线性动作编辑器）操作界面

在 Blender 中，所有的属性和参数都能被制作成动画，这一切都归功于 Blender 使用的一种 RNA 数据结构，它令数据模块之间的链接更加灵活方便。NLA 的具体使用方法将在第 16 章深入讨论。

#### 2.3.5.1 动画模块之间的关系

前面曾讲过，Action（动作）块是由多个 F-Curve（曲线）控制的基本属性动画模块，那么 Dope Sheet 就是将多个 Action 模块组合成为一个 Action（动作）体，而在 NLA 中每个基本编辑元素都是一个 Action 体，因此 NLA 可以被称为 Blender 动画编辑中的最顶层工具。如图 2-75 所示为 Blender 各动画模块之间的关系图。

为了解释动画模块之间的关系和各工具在不同编辑层次上的应用，下面使用一个简单的例

子详细说明。

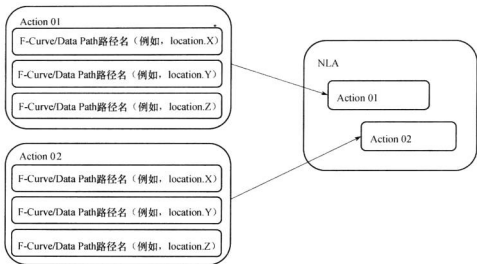


图 2-75 Blender 各动画模块之间的关系

例如需要制作一个完整的步行动画，其中指尖的 Z 轴坐标就由一个 F-Curve 属性曲线来控制。首先在 Graph Editor 中对这个曲线做精确的编辑与定位，控制某一个瞬间其属性值的变化，也就是脚尖此刻在 Z 轴上的坐标位置。同时脚尖的其他属性，例如 X 轴和 Y 轴上的坐标数据，都将以通道的方式分别在 Graph Editor 上表示出来。

脚尖在这个瞬间的动作数据就是由多个 F-Curve 封装成的一个 Action（动作）块，这个动作块包括指尖的 Z 轴和 Y 轴上的坐标值等，这一切在 Dope Sheet 中将被表示为一个小菱形。脚尖从离地抬至最高位置到完成一个完整的迈步行走动作，需要多个 Action（动作）块在时间轴上配合组成，这一系列动作变化称为 Action（动作）体，在 Dope Sheet 中将被表示为一组小菱形组。不同物体（例如，不同骨骼）在同一时间上的动作块将以通道的方式在 Dope Sheet 上分别显示出来。

可以分别制作出行走的动作体和跑步的动作体，然后将这两个 Action（动作体）导入至 NLA 中，使用自动插值原理计算出两个动作之间的过渡动作体，这样就十分轻松地生成一套人物从行走到跑步的完整 Action（动作）。动作体在 NLA 中的编辑和 Video Sequence Editor（视频编辑器）中对视频素材处理的方法类似，每一个动作体都将按照时间长度显示在 NLA 的时间轴上。由于 NLA 的这种非线性特性，它允许重复调用动作体素材，减少了对同一类型动作的多次重复制作。NLA 中的通道表示了当前物体的动作体编辑轴，可以在一个通道上添加多个子动作体，例如放置行走动作体和跑步动作体，以及自动生成的过渡动作体，而摄像机的动作体数据则将显示在同一个时间点上的另一个通道中。

#### 2.3.5.2 View（视图）菜单

如图 2-76 所示为 NLA（非线性动画编辑器）窗口中的 View（视图）菜单。

NLA 的 View（视图）菜单同 Graph Editor（图形编辑器）类似，请参考前文对后者的说明。在窗口中单击快捷键 N，可调出 NLA 的动作

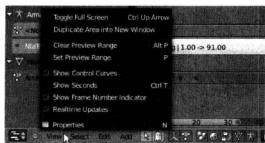


图 2-76 View（视图）菜单

体属性框,如图 2-77 所示。

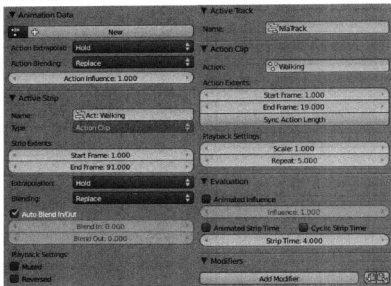


图 2-77 NLA 动作体属性

**Animation Data:** 用于定义当前动作轴的通用属性。

**Active Strip:** 定义当前选择的动作体属性,包括命名、类型和播放帧数等。

**Active Track:** 定义了当前激活的通道名称。

**Action Clip:** 设置当前激活的动作体基本属性,例如重复播放的次数等。

**Evaluation:** 动作体的评估属性。

**Modifiers:** 与 F-Curve (曲线) 中的修改器类似,可以借助不同类型的修改器制作特殊的动作体。

### 2.3.5.3 Select (选择) 菜单

如图 2-78 所示为 NLA (非线性动画编辑器) 窗口中的 Select (选择) 菜单。

NLA 的 Select (选择) 菜单同 Graph Editor (图形编辑器) 类似,请参考前文对后者的说明。

### 2.3.5.4 Edit (编辑) 菜单

如图 2-79 所示为 NLA (非线性动画编辑器) 窗口中的 Edit (编辑) 菜单。



图 2-78 Select (选择) 菜单

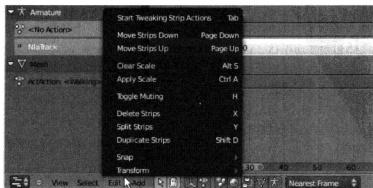


图 2-79 Edit (编辑) 菜单

**Start Tweaking Strip Actions:** 在 Dope Sheet 编辑页面中打开当前选择的动作体, 快捷键为 Tab。

**Move Strips Down/Up:** 在通道中移动 Action (动作) 体, 快捷键为 Page Down 和 Page Up。

**Clear Scale:** 清除对动作体的缩放修改, 快捷键为 Alt + S。因为这里缩小一个动作体, 相当于快放了当前的动作。

**Apply Scale:** 应用对动作体的缩放, 快捷键为 Ctrl + A。

**Toggle Muting:** 暂停当前的动作效果, 快捷键为 H。

**Delete Strips:** 删除动作体, 快捷键为 X。

**Split Strips:** 切割动作体, 快捷键为 Y, 切割点为鼠标指向的位置。

**Duplicate Strips:** 复制动作体, 快捷键为 Shift + D。

**Snap:** 启用吸附操作功能。

**Transform:** 常用的动作体控制选项, 包括移动和缩放等, 快捷键为 G 和 S。

### 2.3.5.5 Add (添加) 菜单

如图 2-80 所示为 NLA (非线性动画编辑器) 窗口中的 Add (添加) 菜单。



图 2-80 Add (添加) 菜单

**Add Tracks Above Selected:** 添加动作体通道。

**Add/Remove Meta-Strips:** 合并或分解当前选择的多个动作体, 如图 2-81 左图所示为合并前的多动作块, 右图为合并后的动作体集。



图 2-81 Meta-String (动作体集)

**Add Transition:** 为两个动作体之间添加过渡动作体, 例如我们分别制作站立动作体和跑步动作体, 可以使用 Transition 功能为两个动作之间自动生成过渡动作体, 如图 2-82 所示, 蓝色素材段为自动生成的过渡动作体。

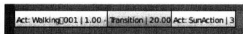


图 2-82 Transition (过渡动作体)

**Add Action Strip:** 添加动作体。

### 2.3.6 图片编辑器

如图 2-83 所示为 UV/Image Editor (图片编辑器) 的窗口。

UV/Image Editor 图片编辑器常用于修改 UV、绘制贴图和调整图片色彩等, 它的初始界面很简单, 当选择不同的模式时, 界面将调用不同功能的选项和菜单。

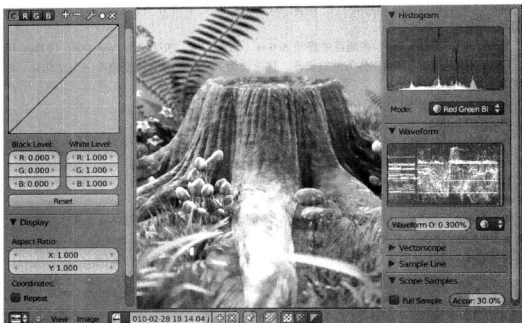


图 2-83 UV/Image Editor (图片编辑器) 编辑窗口 (图片来自电影《Big Buck Bunny》)

### 2.3.6.1 View (视图) 菜单

如图 2-84 所示为 UV/Image Editor (图片编辑器) 窗口中的 View (视图) 菜单。

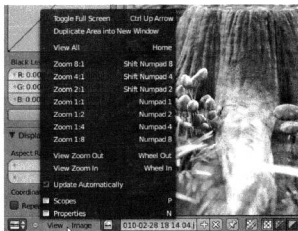


图 2-84 View (视图) 菜单

**Toggle Full Screen:** 窗口最大化, 快捷键为  $\text{Ctrl} + \uparrow$ 。

**Duplicate Area into New Window:** 复制当前窗口到一个新的窗口。

**View All:** 自适应缩放可查看全部内容, 快捷键为 Home。

**Zoom:** 缩放操作, 快捷键为 MMB。

**Scopes:** 弹出色彩范围图, 快捷键为 P。

**Properties:** 弹出属性菜单, 快捷键为 N, 可查看当前图片的 Histogram (直方图) 等色彩参数。

### 2.3.6.2 Image (图片) 菜单

如图 2-85 所示为 UV/Image Editor (图片编辑器) 窗口中的 Image (图片) 菜单。

**Image Painting:** 开启绘制功能。

**Pack:** 将图片打包到工程文件中。

**Edit Externally:** 使用外置软件处理当前图片。

**Save As:** 另存为, 快捷键为 F3。

**Save:** 保存, 快捷键为 Alt + S。

**Reload:** 重载图片, 快捷键为 Alt + R。

**Replace:** 替换当前图片。

**Open:** 打开图片, 快捷键为 Alt + O。

**New:** 新建图片, 快捷键为 Alt + N。

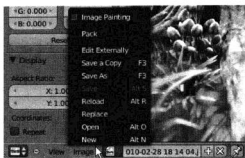


图 2-85 Image (图片) 菜单

### 2.3.7 视频序列编辑器

如图 2-86 所示为 Video Sequence Editor (视频序列编辑器) 的窗口。

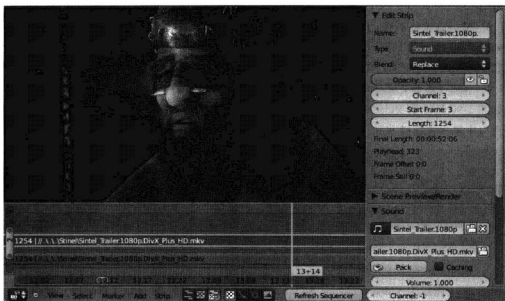


图 2-86 Video Sequence Editor (视频序列编辑器) 编辑窗口 (截图来自电影《Sintel》)

VSE 是一款用于视频后期处理的非线性编辑器, 编辑元素为视频、音频和图片文件, 并支持特效添加。具体使用方法将在第 10 章中深入讨论。

#### 2.3.7.1 View (视图) 菜单

如图 2-87 所示为 Video Sequence Editor (视频序列编辑器) 的 View (视图) 菜单。

**Transform Markers:** 允许编辑 Marker 时间轴的帧标识符。

**Safe Margin:** 启用安全框口查看模式。

**Show Frame Number Indicator:** 在时间标识上显示帧数指示器。

**Draw Frames:** 在标识指示器上使用帧数显示, 而不是时间秒数显示。

**View Selected:** 显示所选。

**Fit preview in window/View all Sequences:** 自动缩放视图, 使当前时间轴全部可见。

## 2.3.7.2 Select (选择) 菜单

如图 2-88 所示为 Video Sequence Editor (视频序列编辑器) 的 Select (选择) 菜单。

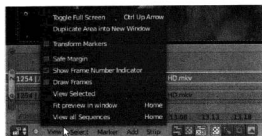


图 2-87 View (视图) 菜单



图 2-88 Select (选择) 菜单

**Right/Left Handle:** 选择当前素材段的左右控制杆。

**Surrounding Handles:** 同时选择素材段左右两端的控制杆。

**Strips to the Right/Left:** 选择当前激活素材段的左右相邻素材段。

## 2.3.7.3 Marker (标识符) 菜单

如图 2-89 所示为 Video Sequence Editor (视频序列编辑器) 的 Marker (标识符) 菜单, 它的使用和 Timeline (时间轴) 中一致, 请参考对后者 Marker 的描述。

## 2.3.7.4 Add (添加) 菜单

如图 2-90 所示为 Video Sequence Editor (视频序列编辑器) 的 Add (添加) 菜单, 也可以直接在窗口中单击快捷键 Shift + A, 弹出添加素材物体的菜单。



图 2-89 Marker (标识符) 菜单



图 2-90 Add (添加) 菜单

## 2.3.7.5 Strip (视频段) 菜单

如图 2-91 所示为 Video Sequence Editor (视频序列编辑器) 的 Strip (视频段) 菜单。

**Swap Strip:** 素材段的相邻左右切换操作, 快捷键为 Alt + 方向键。

**Snap Strips:** 启用吸附编辑, 快捷键为 Shift + S。

**Mute Deselected Strips:** 静默未选择的素材段, 快捷键为 Shift + H。

**Un-Mute/Mute Strips:** 使当前选择的素材段静默, 快捷键为 H, 取消静默的快捷键为 Alt + H。

**UnLock/Lock Strips:** 解锁/锁定当前素材段, 使其可被/无法被编辑, 锁定的快捷键为 Shift + L, 解锁的快捷键为 Shift + Alt + L。

**Reassign Inputs/Reload Strips:** 重置当前输入的素材段。

**UnMeta/Make Meta Strip:** 将素材段群组成集, 和 NLA 中用法类似, 快捷键为 Alt + M 和 M。

**Deinterlace Movies:** 影片帧速控制器。

**Separate Images:** 对图片类素材段做分割操作, 快捷键为 Y。

**Cut (soft/hard) at frame:** 切割视频素材段, 快捷键为 K。



**Grab/Move/Extend from frame:** 延长当前的素材片段, 快捷键为 E。

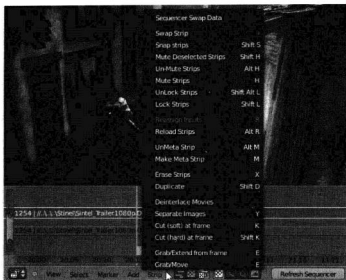


图 2-91 Strip (视频段) 菜单

### 2.3.8 文字编辑器

Text Editor (文字编辑器) 可用于书写文档、编写脚本程序, 同时也是一款功能强大的 Python IDE 开发环境。如图 2-92 所示为 Text Editor (文字编辑器) 的界面。

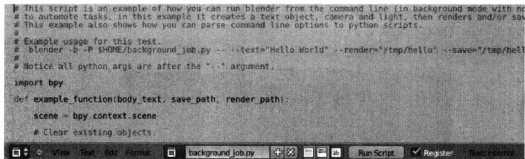


图 2-92 Text Editor (文字编辑器) 编辑窗口

#### 2.3.8.1 View (视图) 菜单

如图 2-93 所示为 Text Editor (文字编辑器) 的 View (视图) 菜单。

**Bottom of File:** 跳跃至文本的底部, 快捷键为 Ctrl + End。

**Top of File:** 跳跃至文本的顶部, 快捷键为 Ctrl + Home。

**Duplicate Area into New Window:** 创建一个新的独立文本编辑窗口。

**Properties:** 调出属性面板, 其功能与标题栏菜单相同, 快捷键为 Ctrl + F。属性中包括了 Line Numbers (行数显示)、Word Wrap (自动换行)、Syntax Highlight (高亮标点符号)、Live Edit (实时编辑)、Font Size (字体大小)、Tab Width (缩进宽度)、Tab as Spaces (与空格同宽)、Find (搜索) 等, 如图 2-94 所示。

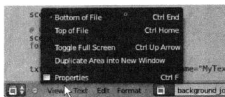


图 2-93 View (视图) 菜单

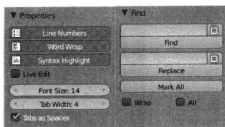


图 2-94 文字 Properties (属性) 面板

### 2.3.8.2 Text (文本) 菜单

如图 2-95 所示为 Text Editor (文字编辑器) 的 Text (文本) 菜单。

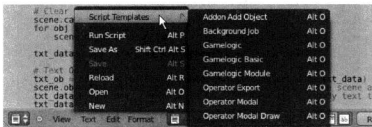


图 2-95 Text (文本) 菜单

**Script Templates:** 脚本模板, 如图 2-95 所示, 文字编辑器中集成了一些基本的脚本模板, 多用于游戏的 Logic (逻辑) 编写。

**Run Script:** 单击后可以运行编辑的脚本, 快捷键为 Alt + P。

**Save As/Save/Reload/Open/New:** 保存、打开和新建等常用的文件功能。

### 2.3.8.3 Edit (编辑) 菜单

如图 2-96 所示为 Text Editor (文字编辑器) 的 Edit (编辑) 菜单。

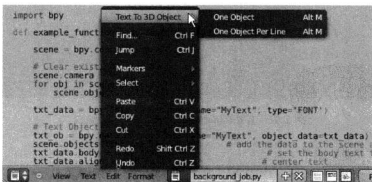


图 2-96 Edit (编辑) 菜单

**Text to 3D Object:** 将编辑文字转换为 3D 物体, 快捷键为 Alt + M。

**Find:** 查找功能, 快捷键为 Ctrl + F。

**Jump:** 快速跳跃至指定行, 快捷键为 Ctrl + J。

**Markers:** 标识符的编辑与查找选项。

**Select:** 选择操作, 例如全选操作的快捷键为 Ctrl + A。

**Paste/Copy/Cut/Redo/Undo:** 粘贴/复制/剪切/重复/撤销操作, 快捷键分别为 Ctrl + V, Ctrl + C, Ctrl + X, Ctrl + Shift + Z 和 Ctrl + Z。

#### 2.3.8.4 Format (格式) 菜单

如图 2-97 所示为 Text Editor (文字编辑器) 的 Format (格式) 菜单。

**Convert Whitespace:** 将空格符转换为制表符。

**Comment/Uncomment:** 将当前选择列作注释和取消注释操作, 快捷键为 Shift + Ctrl + D。

**Indent/Unindent:** 对当前选择列作缩进操作, 快捷键为 Tab 和 Shift + Tab。

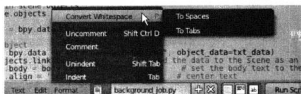


图 2-97 Format (格式) 菜单

#### 2.3.9 结点编辑器

结点编辑器可对材质、纹理和渲染结果添加更多复杂的效果, 具体使用将在第 10 章中做更深入的讨论。如图 2-98 所示为 Node Editor (结点编辑器) 的界面。



图 2-98 Node Editor (结点编辑器) 编辑窗口

##### 2.3.9.1 View (视图) 菜单

如图 2-99 所示, View (视图) 菜单提供了一些常用的缩放和属性工具。



图 2-99 View (视图) 菜单

**View All:** 显示当前窗口中的所有结点, 快捷键为 Home。

**Properties:** 显示属性面板, 快捷键为 N。

##### 2.3.9.2 Select (选择) 菜单

如图 2-100 所示为 Node Editor (结点编辑器) 的 Select (选择) 菜单。

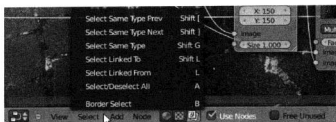


图 2-100 Select (选择) 菜单

**Select Same Type Prev/Next:** 选择相邻的同类型结点, 快捷键为 Shift + [ 或 Shift + ]。

**Select Linked To:** 选择当前激活结点的输出结点, 快捷键为 Shift + L。

**Select Linked From:** 选择当前结点的输入结点, 快捷键为 L。

**Select/Deselect All:** 全选或取消全选, 快捷键为 A。

**Border Select:** 框选操作, 快捷键为 B。

#### 2.3.9.3 Add (添加) 菜单

如图 2-101 所示为 Node Editor (结点编辑器) 的 Add (添加) 菜单, 用于添加不同类型的结点。

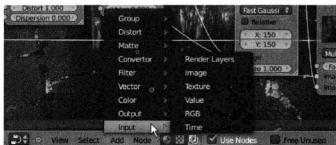


图 2-101 Add (添加) 菜单

#### 2.3.9.4 Node (结点) 菜单

Node (结点) 菜单提供了结点系统中的所有操作工具选项, 如图 2-102 所示。



图 2-102 Node (结点) 菜单

**Show Cyclic Dependencies:** 显示独立的循环结构, 快捷键为 C。

**Toggle Hidden Node Sockets:** 将结点的参数面板收起, 快捷键为 Ctrl + H。

**Toggle Node Preview:** 将结点的预览面板收起, 快捷键为 Shift + H。

**Toggle Node Mute:** 屏蔽结点的功能效果, 快捷键为 M。

**Hide:** 隐藏当前选择的结点, 快捷键为 H。

**Group/Ungroup:** 为选择的结点建立群组, 快捷键为 Ctrl + G, 解除群组的快捷键为 Alt + G。

**Edit Group:** 编辑选择的群组, 快捷键为 Tab。

**Make and Replace Links:** 建立或者替换链接, 快捷键为 Ctrl + F。

**Make Links:** 在选择的两个结点之间建立连接关系, 快捷键为 F。

**Delete:** 删除结点, 快捷键为 X。

**Duplicate:** 复制结点, 快捷键为 Shift + D。

**Resize/Rotate/Translate:** 缩放、旋转、移动操作, 快捷键为 S/R/G。

### 2.3.10 逻辑编辑器

如图 2-103 所示为 Logic Editor (逻辑编辑器) 的主界面, 用于游戏的逻辑编写开发。具体使用将在第 12 章中重点讨论。

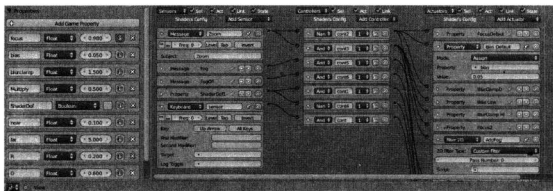


图 2-103 Logic Editor (逻辑编辑器) 编辑窗口

### 2.3.11 工具面板

Properties (工具) 面板包括了所有的工具菜单和属性选项。如图 2-104 所示为工具面板上各图标所标注的功能子面板, 使用 LMB 做切换操作。每个面板的具体使用将在后面章节中详细介绍。

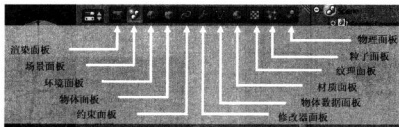


图 2-104 Properties (工具) 面板

### 2.3.12 大纲管理器

Outline (大纲) 管理器类似于操作系统中的资源管理器, 它使用树形结构图来显示场景中

物体间的层次关系，使用户可以更快地定位场景中的物体和其属性，更直观地处理物体与属性之间的关系。如图 2-105 所示为 Outline（大纲）管理器的窗口。

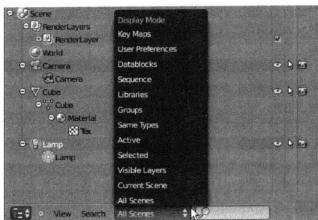






图 2-105 Outline（大纲）管理器窗口

大纲管理器有多种视图模式，不同的模式分别用于显示不同属性物体之间的关系列表，例如 Key Maps（快捷键列表）、User Preferences（参数）列表、Datablocks（数据模块）列表、Sequence（序列）列表、Libraries（库）列表、Groups（组）列表、Same Types（同类型物体）列表、Active（当前激活）物体的属性列表、Selected（被选择）物体的属性列表、Visible Layers（可见图层）的物体列表、Current Scene（当前场景）物体的属性列表、All Scenes（全部物体）的属性列表。

在大纲管理器中，使用 LMB 来选择单独的列对象。如图 2-106 所示，在物体上单击 RMB 会弹出一个物体操作方式的菜单，包括 Select（选择）、Deselect（取消选择）、Delete（删除）、Toggle Visible（使物体不可见）、Toggle Selectable（使物体锁定不可选）、Toggle Renderable（使物体不被渲染）。对于最后的三个功能可以分别使用对象旁的三个小图标来快速操作：小眼睛图标  睁开代表物体可见，单击后小眼睛  闭上则代表物体被隐藏；同理，旁边的箭头  图标代表物体是否允许被选择编辑； 相机图标代表渲染时物体是否可见。

大纲管理器还有一个强大的功能——可以选择对象，并在场景中直接做拖拽添加和属性复用的操作。如图 2-107 所示，在物体模式下选择列表中的 Cube 物体，单击 LMB 按住 Cube 不放，然后往场景中直接拖拽，Blender 会自动新建一个新的 Cube 复制对象出来，同时新的复制物体会被添加一个“.00X”的后缀。也可以对物体的属性做拖拽应用，例如，将一个材质拖拽到物体上进行快速的属性复用。

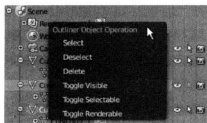


图 2-106 物体操作菜单

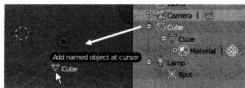


图 2-107 拖拽功能

### 2.3.13 信息窗口

Info（信息）窗口面板提供了系统的主要通用功能和资源信息。如图 2-108 上图所示，分别


为布局和场景选择菜单。如图 2-108 下图所示的 Blender Render 菜单包含了系统可供使用的渲染器，单击下拉菜单可从中选择所需要的渲染设置。标题栏右边的 Ve 表示了当前场景中物体的点数量，Fa 表示了面数量，Ob: 1-4 表示场景中总共有 4 个物体，有 1 个处于选择激活状态，Mem: 4.52M (0.76M) 表示当前的内存使用情况，Cube 则表示当前选择的物体名称。最右边的小箭头  图标可切换画面至全屏操作，快捷键为 Alt + F11。



图 2-108 Info (信息) 面板

### 2.3.13.1 File (文件) 菜单

如图 2-109 所示为 Info (信息) 窗口面板的 File (文件) 菜单。



图 2-109 File (文件) 菜单

**New:** 新建一个工程，快捷键为 Ctrl + N。

**Open/Open Recent:** 打开一个工程、打开最近建立的工程，快捷键为 Ctrl + O。

**Recover Last Session/Recover Auto Save:** 恢复最近一次的工程、打开自动保存的工程文件。

**Save/Save As/Save Copy:** 保存、另存为和保存副本选项，快捷键分别为 Ctrl + S、Shift + Ctrl + S 和 Ctrl + Alt + S。

**User Preferences/Load Factory Settings:** 用户设置界面，快捷键为 Ctrl + Alt + U；初始化设置至出厂模式，快捷键为 Ctrl + N。

**Link:** 用于在当前场景中 Link 链接其他工程文件，将物体导入使用但是禁止修改，快捷键为 Ctrl + Alt + O。

**Append:** 直接导入其他工程文件中的物体，并允许对其做编辑等操作，快捷键为 Shift + F1。

**Import/Export:** Blender 兼容 3DS、DXF、OBJ、VRML 等文件的导出和导入应用。

**External Data:** 修改工程文件中的打包文件，例如将贴图和运算数据等保存在工程中。

**Quit:** 退出，快捷键为 Ctrl + Q。注意：Blender 不会提示你是否确定退出，所以切记谨慎使用这个快捷键。

### 2.3.13.2 Add (添加) 菜单

Add 菜单用于在场景中添加新的物体，这里的分类清晰明确。如图 2-110 所示，每种类别的菜

单下都有添加物体的二级选项菜单，新添加的物体将出现在场景中的光标位置，快捷键为 Shift + A。



图 2-110 Add (添加) 物体菜单

### 2.3.13.3 Render (渲染) 菜单

如图 2-111 所示为 Info (信息) 窗口面板的 Render (渲染) 菜单。

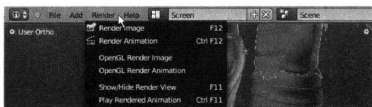


图 2-111 Render (渲染) 菜单

**Render Image/Animation:** 渲染当前场景或动画，快捷键分别为 F12 和 Ctrl + F12。

**OpenGL Render Image/Animation:** 快速渲染 OpenGL 图像或动画。

**Show/Hide Render View:** 显示或隐藏渲染窗口，快捷键为 F11。

**Play Rendered Animation:** 动画回放，快捷键为 Ctrl + F11。

### 2.3.13.4 Help (帮助) 菜单

Help (帮助) 菜单提供了常用的帮助文档和信息链接，请读者自行参考软件列表。

## 2.3.14 文件浏览器

当单击 F1 使用打开或浏览文件等功能时，Blender 都会自动启动 File Browser (文件浏览器)，用于显示系统中的文件列表。如图 2-112 所示为 File Browser (文件浏览器) 的窗口界面。

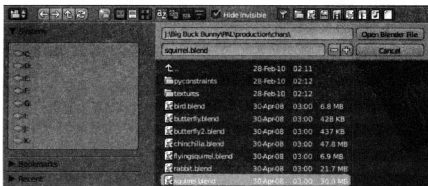




图 2-112 File Browser (文件浏览器) 窗口



文件浏览器的操作和管理方法与操作系统的资源管理器类似，其中  按钮组可以使文件列表按照一定的顺序来排列，例如按字母、后缀名和文件大小等。也可以选择列表、详细列表和缩略图等不同的浏览方式。过滤器按钮组  是一个文件类型过滤器，当开启后可以选择过滤掉一些不需要查看的文件，十分方便。

### 2.3.15 控制台窗口

Console（控制台）窗口，用于显示系统的运行状况，同时也用于代码编译和调试，实时查看游戏脚本和逻辑运行结果等。如图 2-113 所示为 Console（控制台）窗口的界面。

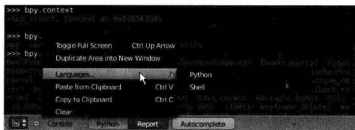


图 2-113 Console（控制台）窗口

**Languages:** 切换命令行的语言输入格式，包括 Python 和 Shell 两种类型。

**Paste from/Copy to Clipboard:** 复制和粘贴操作，快捷键分别为 Ctrl + C 和 Ctrl + V。

**Clear:** 清除当前界面中的所有输入。

## 第3章

# 建模

Modeling（建模）是在3D视图环境中利用点线面等工具创建出三维模型的过程，同时，建模也是整个CG（Computer Graphics 计算机制图）制作流程中最基础的步骤。养成一个好的操作习惯，并在制作的过程中保持清晰的布线思路，有助于快速而高效地完成建模工作，得到满足工程需求的模型。

Blender 提供了十分强大的建模工具集，包括常用的基本多边形工具和 NURBS 曲线高级建模工具等。对于初级的三维创建，主要使用默认的几何体，通过基本的挤压、切割和变形等操作方法，制作出复合造型。

模型的质量决定了物体的最终视图效果，同时也影响了其应用性能。一个高质量的模型能够记录合理的设计数据，使模型在较少的多边形数目下，可以表现出非常复杂的造型，如图3-1所示。

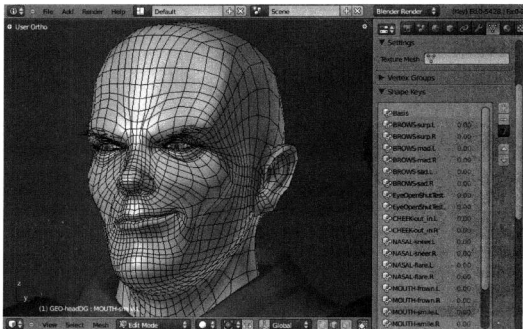


图 3-1 高性能的模型结构可以表现复杂的动作细节

### 3.1 新建物体

Blender 提供了十分丰富的基本物体，可以将其直接添加至场景中，通过修改操作来制作更复杂的物体模型。首先在 Info（信息）标题栏的 Add 菜单中，可以找到所有基本物体的分类添加菜单，如图 3-2 所示。也可以在 3D 窗口中使用快捷键 Shift + A，弹出添加物体的快捷菜单。



图 3-2 默认物体添加菜单

#### 3.1.1 网格物体

Mesh（网格）菜单包括了常用的网格物体，如图 3-3 所示，图中为常用添加的网格物体。

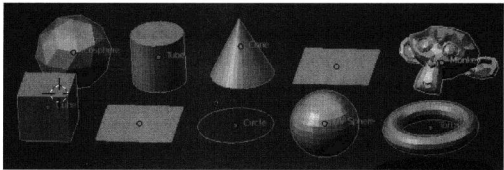


图 3-3 常用的 Mesh 物体

图中新建的 Plane 物体和 Grid 物体在物体模式下造型完全相同，它们之间唯一的区别就是，Grid 是一个网格数为  $10 \times 10$  的 Plane 物体。单击 Tab 进入它们的编辑模式，即可看到两者的区别。如图 3-4 所示，左图为 Plane 物体，右图则是 Grid 物体。

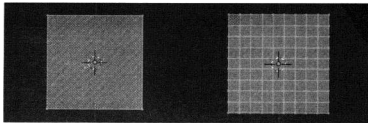


图 3-4 Plane 物体和 Grid 物体的区别

Circle (圆) 是一个可自定义节点数的封闭线段, 在添加的时候可通过工具面板自定义其节点的数量, 当线段的节点数较多时, Circle 物体就会近似为一个圆形圈。如图 3-5 所示, 从左至右分别是 32 个节点 (默认)、8 个节点和 3 个节点的 Circle (圆) 物体, 其中 Radius (半径) 可初始化圆的半径大小, 也可以单击 Fill (填充) 按钮来将新建物体的表面自动填充起来。

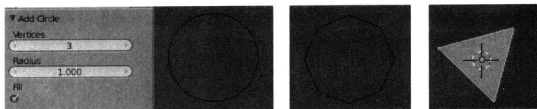


图 3-5 Circle (圆)

对于 UV Sphere (UV 球体), 也可以自定义其初始化状态。如图 3-6 所示, 从左至右分别是默认 32 瓣、12 瓣和 3 瓣的球体, 随着瓣值的降低, 3 瓣球已经完全失去了球体的球形特征。

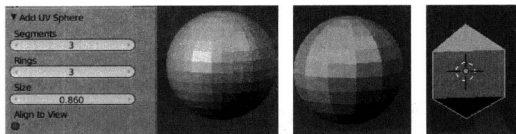


图 3-6 UV Sphere (UV 球体)

不同于 UV Sphere (UV 球体), Icosphere (Ico 球体) 的表面默认由 20 个三角形构成, 初始化的参数为球体表面细分等级。如图 3-7 所示, 从左至右分别是细分等级为 1 级、2 级和 0 级的细分效果。可以将球体最高细分至 500 级, 但是系统可能会因如此高密度的网格而出现崩溃。因为当细分达到 10 级时, 球体将可能产生多达 5 242 880 个三角形, 所以请慎用过高的细分等级参数。

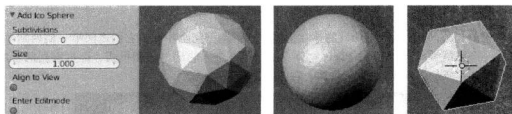


图 3-7 Icosphere (Ico 球体)

### 3.1.2 曲线物体

Curve (曲线) 菜单包括常用的 Bezier 贝济埃曲线、NURBS 曲线和 Path (路径), 如图 3-8 所示。通常使用曲线物体来制作物体的运动路径, 以及放样物体等。

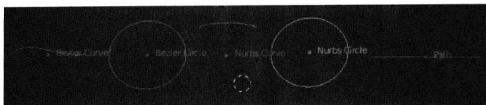


图 3-8 Curve (曲线) 物体

### 3.1.3 骨骼/空物体/摄像机物体

Armature (骨骼) 菜单提供了默认的标准人体骨骼, 以及一些特定部位的关节骨骼。Empty (空) 物体是一个无法编辑的坐标箭头, 常被用于定位和对象控制等。Camera (摄像机) 是渲染画面的取景窗口, 它在 3D 环境中的摆放位置决定了渲染的视角。

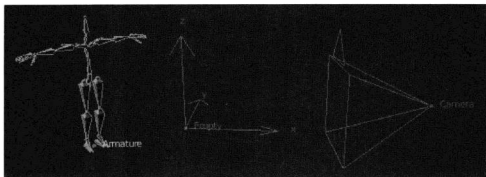


图 3-9 骨骼、空物体和摄像机物体

### 3.1.4 灯光物体

Lamp (灯光) 菜单包含了所有类型的灯光物体, 如 Point (点光源灯)、Sun (日光)、Spot (聚光灯)、Hemi (半球灯)、Area (面光灯), 如图 3-10 所示。

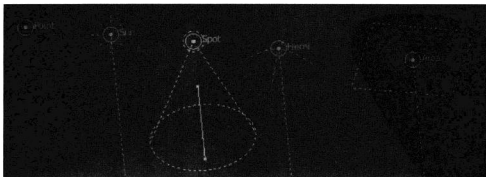


图 3-10 Lamp (灯光) 类型

这里只介绍了一些常用的默认物体, 其他新建类型的物体还包括 Surface (曲面)、Metaball (变形球)、Text (文字)、Lattice (晶格)、Force Field (力场)、Group Instance (实例组) 等。

## 3.2 编辑模式

Mesh (网格) 也称做 Geometry Polygon (几何多边形), 是模型 Object (物体) 的基础属性之一。多边形物体是最常用的建模基本物体, 同时, 多边形建模也是最简单、易于入门的建模学习方式。

Edit Mode (编辑模式) 是在选取 Object (物体) 后, 对其网格属性执行编辑操作的模式。单击鼠标右键选择所需要编辑的物体, 然后在 Mode 面板中选择 Edit Mode, 即可进入该物体的编辑模式, 也可使用快捷键 Tab 进入。如图 3-11 所示, 为物体的编辑模式界面。

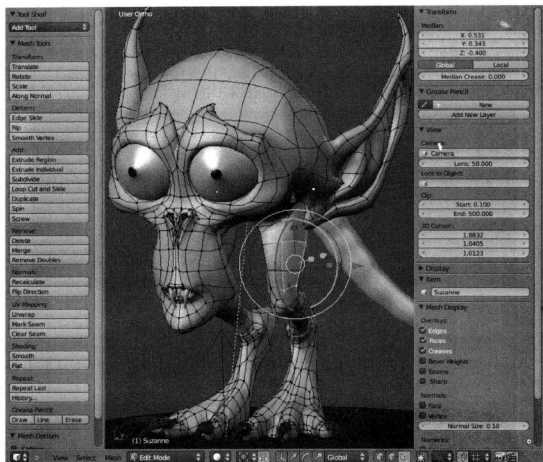


图 3-11 Edit Mode (编辑模式) 界面

对比 Object (物体) 模式, 编辑模式在标题栏处取消了层的选项, 同时 Select (选择) 菜单更新了更多的功能, Object 菜单也被 Mesh (网格) 菜单替代。在窗口中单击快捷键 T 可弹出左侧的工具列表, 这里列举了部分基础的建模工具。单击快捷键 N 可在窗口右侧弹出属性菜单, 这里列举了常用的显示设置和物体属性等。

### 3.2.1 点线面结构

所有的 Mesh (网格) 结构都是由点、线和面三种基本的元素组成, 它和曲线物体的结构不一样。Vertices (点) 是网格的最基础元素, 一个单独的点甚至不能被渲染出来。这里要区别点和 Object (物体) 的 Origin (原心) 概念, 原心不是一个网格点, 它是用来表示物体的存在, 以及世界坐标系中物体初始坐标位置的一个属性点。从视觉上看, 它比普通的 Vertices (点) 稍大, 但是无

法像点一样被选择，如图 3-12 所示。一个物体都必须有一个原心，但是一个物体可以不需要拥有任何点。在 3D 环境中添加一个点十分方便，首先必须切换至物体的编辑模式，使用快捷键 Ctrl + LMB 在任意位置单击即可，再次添加时，新的点会自动和前一个点组成一条 Edges（边）。




图 3-12 Origin（原心）与 Vertices（点）

Edges（边）又称为线，通常是由两个点连接而形成的线段。一条线段也不能被渲染出来，但是当多条线段通过缝合连接在一起时，就可以形成 Faces（面）。Faces（面）是多边形网格中最小可渲染的单元，同时面也组成了多边形网格的物体表面。

根据边数的不同，一个面可以分为 Triangles（三角形）、Quadrangles（四边形）以及 BMesh（多边形）三种类型。应该根据不同的应用场合选择合适的面结构。例如，游戏中就应多使用三角形，因为它的表面平整而且易于计算，不会出现交错的破面。四边形并不是任何情况下都能合理地 Blender 中显示出来，因为规范的四边形只能在夹角都小于 180 度的情况下正常显示。如图 3-13 所示，当所选夹角大于 180 度时，夹角外侧会出现古怪的黑色重叠效果，这样的面结构是不规范的，需要避免使用。



图 3-13 正确使用四边形

编辑模式为网格提供了点、线和面三种选择方式，单击  按钮可以切换不同的选择操作模式，也可以使用 Shift + LMB 同时激活多种选择模式。在 3D 视图中，直接使用快捷键 Ctrl + Tab 也可以快速地切换选择模式。

不同选择模式下的控制杆显示位置也会稍有不同。点模式下的控制杆会置于点中心的位置，如图 3-14 左图所示。线模式下的控制杆会置于线段中心的位置，如图 3-14 中图所示。而面模式下的控制杆会置于多边形的面中心位置，如图 3-14 右图所示。

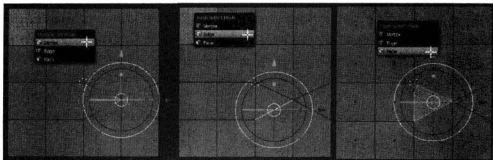


图 3-14 点、线和面的三种选择方式

### 3.2.2 法线结构

Normal (法线) 不是一种实体线, 它是一条垂直于曲线上节点和表面切线方向上的虚拟线, 使用正负值来表示和区别多边形上面和点的法向参数, 并且只有法线为正值的点和面在场景中才可见。一般来说, 由内部指向外部表示当前位置的法线为正向, 如果将表面翻转, 也就改变了法线的方向。如果你的模型不是 Double Sided (双面) 显示的话, 那么负法向的那一面将会在视图窗口中不可见, 并且不可被渲染。

如图 3-15 所示, 在 3D 视图窗口中单击快捷键 N 调出显示面板, 在 Normals (法线) 面板中选中 Face (面) 和 Vertex (点) 选项后, 物体就会在面和点的法向上分别显示法线的方向。左边模型的法线完全一致, 而右边的多边形由于部分面的法线被翻转了, 所以无法在表面上看到负向法线值的面, 同时这一块区域的多边形也由于无法正常显示变为黑色。

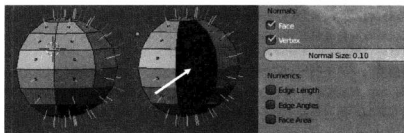


图 3-15 Normals (法线) 显示面板

### 3.2.3 选择菜单

如图 3-16 所示, 为 Edit (编辑) 模式下的 Select (选择) 菜单, 它提供了物体选择操作的所有功能。

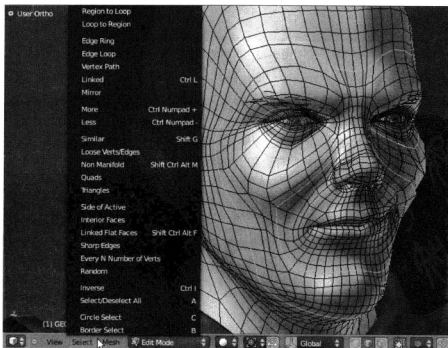


图 3-16 Select (选择) 菜单



### 3.2.3.1 常规选择模式菜单

**Region to Loop/Loop to Region:** 相邻面与边的相互切换选择模式。如图 3-17 所示, 当选择两条连续的线条时, 单击 Loop to Region 可自动选择它们之间相邻范围内的所有面, 也可以使用快捷键 Ctrl + E 调出边选择菜单, 右图为选择后的效果。同样, 在一个面选择的情况下, 使用 Region to Loop 功能可快速地选择当前面的外周边, 如图 3-18 所示。

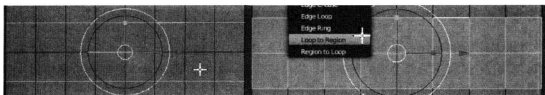


图 3-17 Loop to Region (由循环边选择面)

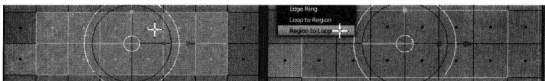


图 3-18 Region to Loop (由面选择循环边)

**Edge Ring/Loop:** 边的循环结构选择, 快捷键为 Ctrl + E。在一个循环面结构中, 当选择任意边时, 单击 Edge Ring, 系统会自动在当前的环面结构中选取相似排列的边, 如图 3-19 所示。而 Edge Loop 可在选择一个边或者面的情况下, 自动选择被选物体所在的全部循环边结构, 快捷键为 Alt + RMB, 效果如图 3-20 所示。

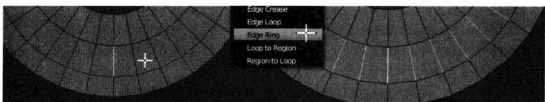


图 3-19 Edge Ring (环形边) 选择

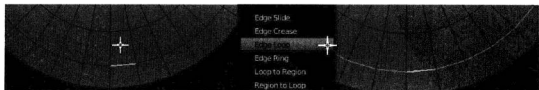


图 3-20 Edge Loop (循环边) 选择

**Vertex Path:** 点路径选择, 它可自动计算并选择两点之间最短距离路径上的所有点。如图 3-21 左图所示, 首先任意选择两个点, 然后单击 Vertex Path 功能, 系统就自动计算并选择了两点之间的最短距离路径线条, 如图 3-21 右图所示。

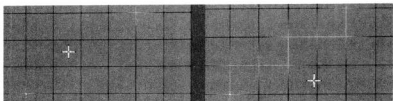


图 3-21 两点间的最短路径选择

**Linked:** 选择被选点所在多边形上的所有连接面，当编辑模式中有多个分离多边形时，可以使用这个功能来选择分离的网格体，快捷键为 L。如图 3-22 所示，在同一个网格物体的编辑模式下，场景中拥有两个分离的网格体，在左边的多边形上任选一个点，单击 L 后系统会自动选择整个相连的网格面，而右边的多边形由于未与左边的网格连接，所以不会被自动选择。

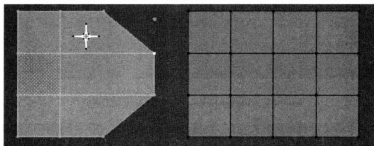


图 3-22 选择（相邻的所有点）

### 3.2.3.2 特殊选择模式菜单

**Mirror:** 镜像选择，选择目标为当前激活点在镜像坐标上的多边形。

**More/Less:** 步进递增或递减地复选被选择点的相邻多面体，快捷键为 Ctrl + Num + / -。

**Similar:** 相似选择，快捷键为 Shift + G，单击后可弹出选择菜单，如图 3-23 所示。可通过这个方法选择属性近似的同类多边形。

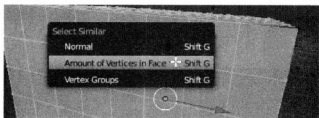


图 3-23 Similar（相似）选择

**Loose Verts/Edges:** 选择场景中没有组成面的点和边。

**Quads/Triangles:** 快速选择场景中的四边形或三角形，这个功能仅工作在面模式下。如图 3-24 左图所示，当单击 Quads 按钮时，系统将自动选择多边形上所有的四边形，而圆心附近的三边形将不会被自动选择。如图 3-24 右图所示，则是单击 Triangles 后系统选择所有三角形的效果。

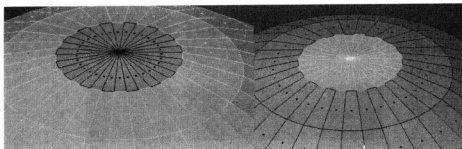


图 3-24 Quads/Triangles (四边形和三角形) 的自动选择方式

**Side of Active:** 自动选择区间上的所有点, 例如 X 正轴方向的所有点。

**Linked Flat Faces:** 选择相连而且夹角相近的所有面, 快捷键为 Shift + Ctrl + Alt + F。如图 3-25 左图所示, 任意选择一个面, 单击 Linked Flat Faces 按钮后, 系统将自动计算选择相连的所有同角度平面。如图 3-25 右图所示为选择的结果。

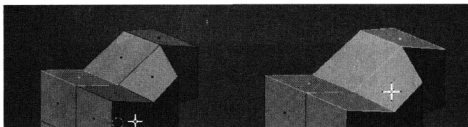


图 3-25 Linked Flat Faces (同角度面) 选择

**Sharp Edges:** 选择所有标识为 Sharp 的边。

**Random:** 让系统做随机选择。

**Inverse/Select/Deselect All:** 反选/全选/取消全选, 快捷键为 Ctrl + I 和 A。

### 3.2.4 网格菜单

如图 3-26 所示, 为 Edit (编辑) 模式下的 Mesh (网格) 菜单, 这里提供了全面的建模操作工具。

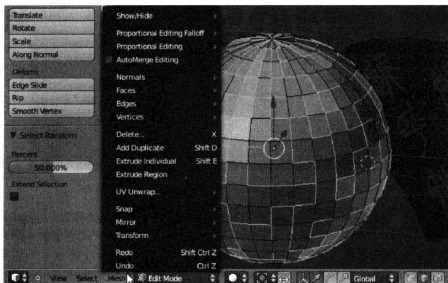


图 3-26 Mesh 网格菜单

### 3.2.4.1 Show Hidden (显示与隐藏) 网格菜单

**Show/Hide:** 如图 3-27 所示, 为隐藏显示的二级菜单。Show Hidden (取消隐藏), 快捷键为 Alt + H。Hide Selected (隐藏选择部分), 快捷键为 H。Hide Unselected (隐藏未选择部分), 快捷键为 Shift + H。



图 3-27 Show/Hide (显示与隐藏) 菜单

隐藏功能常用于复杂模型的建模, 或者高面数模型的雕刻过程。隐藏不需要的多边形可避免视觉上的遮挡, 便于观察编辑对象, 同时节约用于物体显示的内存。例如如图 3-28 左图所示, 由于牙齿和嘴唇的遮挡, 很难观察并且编辑口腔内部的结构。所以可以选中牙齿和嘴唇部分, 单击快捷键 H 将它们隐藏起来, 这样就可以方便直观地编辑口腔内部了, 如图 3-28 右图所示。视图隐藏并不影响网格在渲染中的显示, 可以使用取消隐藏功能来恢复视图中的显示。

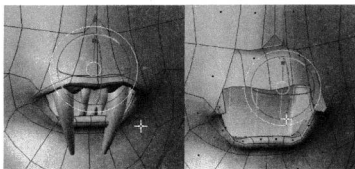


图 3-28 Show/Hide (显示与隐藏) 被选择的网格

### 3.2.4.2 Proportional Falloff (衰减) 编辑模式菜单

**Proportional Editing Falloff:** 衰减编辑模式, 功能同标题栏上的按钮, 如图 3-29 所示。

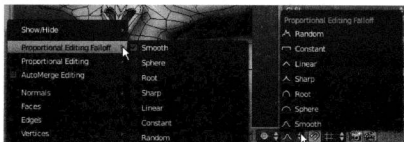


图 3-29 Proportional Editing Falloff (衰减编辑) 模式菜单

这里提供了 7 种衰减选择模式, 分别是 Smooth (平滑)、Sphere (球形)、Root (根部)、Sharp (尖锐)、Linear (线性)、Constant (约束) 和 Random (随机), 菜单中的小图标也形象地说明了该模式在网格表面产生的形变效果。在正常模式下, 选择一个点或者面做移动等形变操作时, 不会对周围的点和多边形造成连带影响。但是当开启比例衰减编辑模式后, 周围

的多边形将会受到衰减模式和衰减范围的影响,产生相应的牵连变化,可以使用 MMB 来缩放控制衰减的影响范围。如图 3-30 所示,可以看到,即使只选择并移动一个点,网格在不同衰减模式下也会呈现不同的附加变形效果,从左至右分别是无衰减正常模式、Smooth、Sphere、Root 衰减模式。如图 3-31 所示,从左至右分别是 Sharp、Linear、Constant 和 Random 衰减模式。

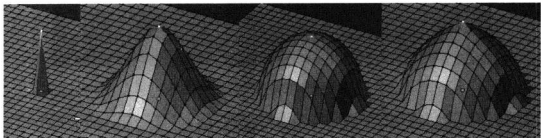


图 3-30 无衰减和 Smooth、Sphere、Root 衰减效果

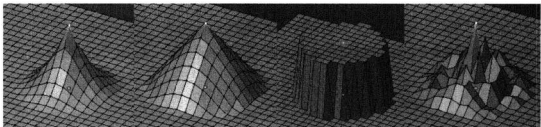


图 3-31 Sharp、Linear、Constant 和 Random 衰减效果

#### 3.2.4.3 Auto Merge (自动缝合) 工具菜单

**AutoMerge Editing:** 自动缝合模式。有些时候边与边之间可能会出现点的重合情况,但是肉眼可能无法区别。重合点会对模型的后期处理,如 UV 解算和动画变形产生一些严重的影响。如图 3-32 左图所示,一个看上去很普通的 6 点矩形,中间却有两个重合点。因为这实际上是两个相邻的正方形,属性显示为 Ve:1-8,说明当前网格上总共有 8 个点,如果未加仔细区分可能会漏掉中间的这个空隙。为了确保没有点重合情况,可以先激活 Auto Merge Editing 功能,再对网格做编辑,这时系统会自动对这些重合的点做合并操作。如图 3-32 右图所示,属性显示为 Ve:0-6,说明这个网格中已经没有重合点了。

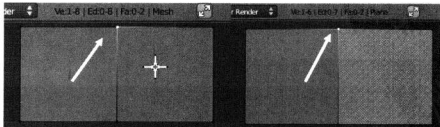


图 3-32 AutoMerge Editing (自动缝合编辑) 模式

#### 3.2.4.4 Normal (法线) 工具菜单

**Normals:** 法线工具,子菜单包括 Recalculate Outside (法线向外),快捷键为 Ctrl + N, Re-

calculate Inside (法线向内), 快捷键为 Shift + Ctrl + N, 以及 Flip Normals (翻转法线)。统一法线是为了能让网格上的相邻面在平滑后都能得到光滑的过渡效果, 这样在渲染时才不会不会出现一些诡异的表面错误。检查法线是确认网格平滑的最基础方式, 如果相邻面的法线没有统一在同一个方向上时, 那么网格在平滑后, 这个相邻位置会出现不平滑的过渡线条, 如图3-33所示。



图 3-33 Normals (法线) 不统一的表面效果

### 3.2.4.5 Vertices (点) 工具菜单

这里列举了所有与点相关的操作方式, 点工具菜单如图3-34所示。

**Merge:** 合并, 首先选择多个需要被合并的点, 使用快捷键 Alt + M, 弹出选项菜单, 如图3-35所示。这里总共有五种合并模式, At First (将多个点合并到第一个选择点的位置), At Last (合并至最后一个选择点的位置), At Center (合并到分布点的中央位置), At Cursor (合并到光标位置) 和 Collapse (折叠合并)。不同的合并模式决定了最后的合并点位置和效果。

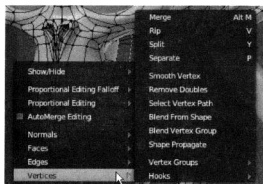


图 3-34 Vertices (点) 工具菜单

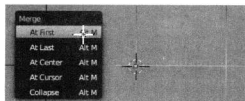


图 3-35 Merge (合并) 模式

**Rip:** 剥离, 将点从所处位置的相邻两个边之间分离开来, 适用于点和线的操作, 快捷键为 V。效果如图3-36所示, 选择一个点, 对其做剥离操作并向左移动, 原接口位置会被自动封闭起来。

**Split:** 分离, 效果和 Rip 类似, 但是只能用于分离面, 快捷键为 Y, 效果如图3-37所示。

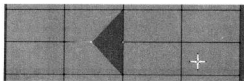


图 3-36 Rip (剥离) 工具

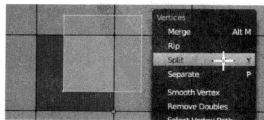


图 3-37 Split (分离) 工具

**Separate:** 将所选择网格分离出当前物体, 创建为一个新的物体, 快捷键为 P, 二级菜单如图3-38所示。

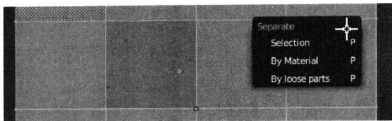


图 3-38 Separate (物体分离) 工具

**Smooth Vertex:** 点平滑功能。

**Remove Doubles:** 移除重合点操作。

**Blend from Shape/Vertex Groups:** 调节点在 Shape (形变帧) 和 Vertex Groups (顶点组) 上的变形过渡。

**Shape Propagate:** 将当前的形变过渡应用至其他形变帧中。

**Vertex Groups:** 顶点组, 详细原理和应用将在后面重点讨论。

**Hooks:** 钩, 和顶点组一样, 用于将一部分点集合起来设置为钩。

#### 3.2.4.6 Edges (边) 工具菜单

这里列举了所有与边相关的操作方式, 如图 3-39 所示, 为边的工具菜单。

**Subdivide:** 细分工具, 当选择一条线条或者面时, 执行细分操作, 每一个线段都会被一分为二。如果这个边刚好处于某个多边形中, 被细分出的连接点会自动与相邻的点缝合成多边形。如图 3-40 所示, 是对一条边执行细分后的多边形重建效果。

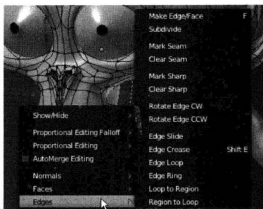


图 3-39 Edges (边) 工具菜单

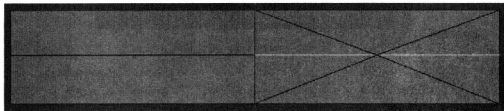


图 3-40 Subdivide (细分) 效果

**Make/Clear Seam:** 制作缝合边, 快捷键为 Ctrl + E, 常用于为 UV 解算做准备。

**Make/Clear Sharp:** 标记或清除硬边。

**Edge Slide:** 边滑动编辑模式, 这是一个十分灵活的边调整工具, 可以在表面上滑动修改被选择边的位置, 而不会影响整个网格的造型。边滑动没有快捷键, 只能单击 Ctrl + E 在边工具菜单中查找, 如图 3-41 左图所示, 调节的效果如图 3-41 右图所示。

**Edge Crease:** 褶皱工具, 用于制作硬边的效果, 快捷键为 Shift + E。

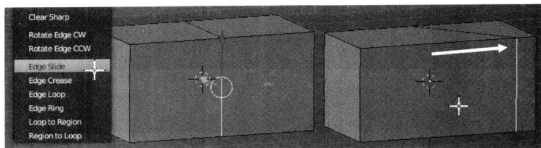


图 3-41 Edge Slide (边滑动) 工具

### 3.2.4.7 Faces (面) 工具菜单

如图 3-42 所示, 为面工具的二级菜单, 这里列举了所有与面相关的操作方式。

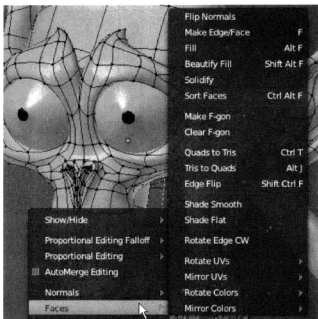


图 3-42 Faces (面) 工具菜单

**Make Edge/Face:** 建立边和面的网格, 快捷键为 F。在几何体中, 两个点能形成一个边, 三个以上的点组成一个面。可以在选择多个点的情况下单击 F, 即可建立新的边和面。如图 3-43 所示, 分别使用 F 将两个点建立一个边, 用 4 个点建立成了一个面。



图 3-43 Make Edge/Face (由点建立边和面)



**Fill/Beautify Fill:** 自动建立完美的缝合面，在多个点的选择基础上，让系统自动生成封闭的面。缝合的原则是使用最少的连接线，所以也因此称为完美缝合，如图 3-44 所示，快捷键分别为 Alt + F 和 Shift + Alt + F。

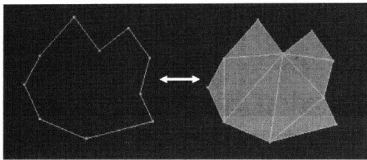


图 3-44 Fill/Beautify Fill (自动建立完美的缝合面)

**Solidify:** 实体化工具，为平面多边形自动添加厚度。如图 3-45 所示，添加一个半圆物体，在全选的情况下单击 Solidify (实体化)，系统会在半圆的内部自动生成稍微小一点的网格圆，完成厚度的实体化操作，可以使用 S 缩放快捷键来调节新建的厚度。

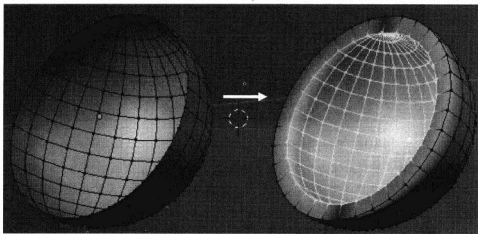


图 3-45 Solidify (实体化) 工具

**Sort Faces:** 重排已选择的网格面，快捷键为 Ctrl + Alt + F。

**Make/Clear F-gon:** 建立 F 封闭面，例如针对如图 3-46 左图所示的多边形，对其执行 Make F-gon 转换操作，网格将隐藏面上的边显示效果，如图 3-46 右图所示。

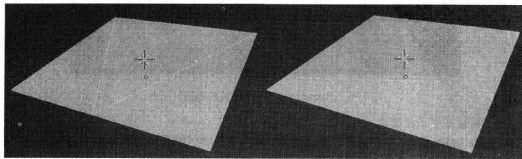


图 3-46 F-gon 网格

**Quads to Tris/Tris to Quads:** 三边形与四边形的互相转换, 快捷键分别是 Ctrl + T 和 Alt + J, 如图 3-47 所示。

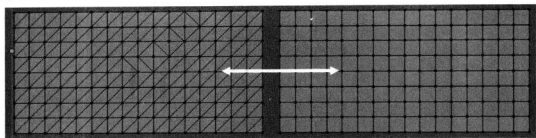


图 3-47 Quads to Tris/Tris to Quads (三边形和四边形) 的互相转换

**Edge Flip:** 边的方向反转工具。

**Shade Smooth/Flat:** 平滑/平坦多边形表面, 转换效果如图 3-48 所示。

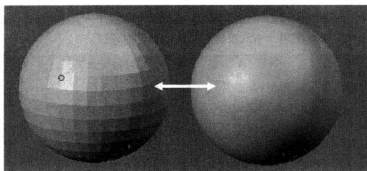


图 3-48 Shade Smooth/Flat (表面平滑与平坦) 的转换工具

**Rotate Edge CW:** 旋转毗邻边, 在选择边时单击 Ctrl + E, 在菜单中使用 CW 做正向旋转, 使用 CCW 做逆向旋转, 如图 3-49 所示, 从左至右分别为正常边结构、CW 正旋一次和 CCW 逆旋一次的效果。

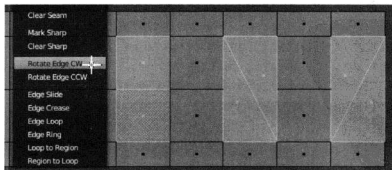


图 3-49 Rotate Edge CW (旋转毗邻边) 工具

### 3.2.4.8 网格形变工具菜单

**Extrude Individual:** 挤压操作, 快捷键为 Shift + E。挤压操作是建模中最常用的命令之一, 用于在原选定多边形的基础上扩展模型。如图 3-50 所示, 分别为对点、边和面做挤压操作的效

果,物体将会在原多边形的基础上向挤压方向扩展。

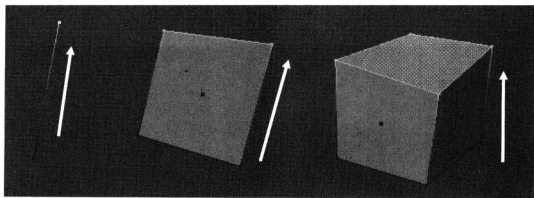


图 3-50 Extrude Individual (挤压) 操作

**Extrude Region:** 在法线方向上做挤压操作。

**UV Unwrap:** UV 解算,用于拆分模型的 UV 图,具体操作将在后面讨论。如图 3-51 所示,为 UV 解算操作的二级菜单。

**Snap:** 吸附功能,同标题栏的磁铁吸附功能。

**Mirror:** 镜像,用于将物体在对称轴上做镜像操作。如图 3-52 所示,为 Mirror (镜像) 的二级菜单。

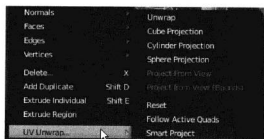


图 3-51 UV Unwrap (UV 解算) 菜单



图 3-52 Mirror (镜像) 菜单

**Transform:** 形变工具,包含了所有基本变形工具。例如,Grab/Move (移动),快捷键为 G; Rotate (旋转),快捷键为 R; Scale (缩放),快捷键为 S; To Sphere (球形化),快捷键为 Shift + Alt + S; Shear (错位),快捷键为 Shift + Ctrl + Alt + S; Warp (弯曲),快捷键为 Shift + W。如图 3-53 所示为 Transform (形变) 工具的二级菜单。

**Grab/Move:** 移动工具,用于移动被选多边形,如图 3-54 左图所示,使用 RMB 选择面,然后单击快捷键 G 即可做随意移动。如果使用 G + X 可将移动约束在 X 轴方向上,如图 3-54 中图所示,而使用 Shift + X + G 组合键可以将移动约束在垂直于 X 轴的 YZ 水平面上,如图 3-54 右图所示。配合 Shift 键可对物体做微移操作,而配合 Ctrl 键可使移动按固定距离做步进移动。

**Rotate:** 旋转工具,用于旋转被选择的多边形。如图 3-55 左图所示,使用 RMB 选择面后单击 R 即可做随意转动。如果使用 R + X 可将旋转约束在围绕 X 轴方向上,如图 3-55

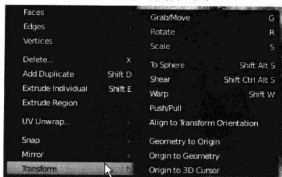


图 3-53 Transform (形变) 工具菜单

中图所示。使用组合键 Shift + Y + R 可以将旋转约束在垂直于 Y 轴的 XZ 水平面上, 如图 3-55 右图所示。和 G 移动操作类似, 配合 Shift 键可以对物体做微旋操作, 配合 Ctrl 键将使移动按固定距离步进旋转。

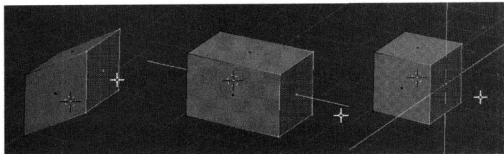


图 3-54 G 移动的组合键操作

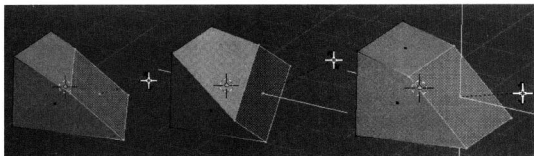


图 3-55 R 旋转的组合键操作

**Scale:** 缩放工具, 用于缩放被选择多边形。如图 3-56 左图所示, 使用 RMB 选择面后单击 S 即可随意缩放。如果使用 S + Y 可将缩放约束在围绕 Y 轴方向上, 如图 3-56 中图所示。Shift + Y + S 组合键则将缩放约束在垂直于 Y 轴的 XZ 水平面上, 如图 3-56 右图所示。使用 Shift + S 可微缩物体, Ctrl + S 可将使移动按固定距离步进缩放。

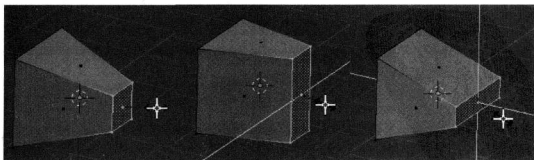


图 3-56 S 缩放的组合键操作

**To Sphere:** 球形化工具, 用于将多边形趋向于球体化变形。如图 3-57 左图所示, 是一个正方形物体。当全选并使用 Shift + Alt + S 球形化操作, 正方形将被变形并趋向于转变成一个球体, 效果如图 3-57 右图所示。物体的细分等级越高, 球形化的效果越好。

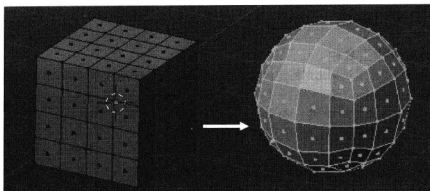


图 3-57 To Sphere (球形化) 工具

**Shear:** 错位工具, 可将多边形在一定方向轴上做错位变化。如图 3-58 左图所示, 为一个初始状态的正方形, 右图是使用 Shift + Ctrl + Alt + S 变化后的效果。

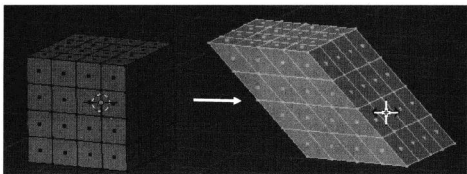


图 3-58 Shear (错位变化) 工具

**Warp:** 弯曲工具, 能将所选多边形以光标为圆心做弯曲变化。如图 3-59 所示, 为将左边的 Cube 物体做弯曲变化后的效果。

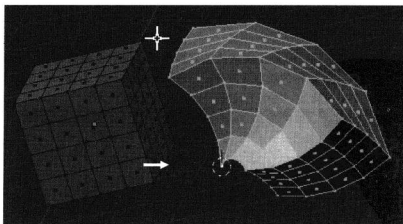


图 3-59 Warp (弯曲变化) 工具

## 3.3 基础网格建模

### 3.3.1 网格快捷工具栏

在3D窗口中单击T可调用网格建模快捷工具栏，用于快速地选择常用的网格相关工具。这里主要包括：Transform（变形类）、Deform（形变类）、Add（添加类）和Remove（移除类）工具等选项，如图3-60所示，为工具栏菜单。

Transform（变形类）工具包含了Translate（移动工具），快捷键为G；Rotate（旋转工具），快捷键为R；Scale（缩放工具），快捷键为S。

Deform（形变类）工具包含了Edge Slide（边滑动工具）、Rip（剥离工具）、Noise（噪点工具）以及Smooth Vertex（点平滑工具）。

Add（添加类）工具包含了常用的网格扩展和修改工具，如Extrude Region/Individual（挤压工具），快捷键为E；Subdivide（细分工具），快捷键为W；Loop Cut and Slide（环切工具），快捷键为Ctrl+R；Duplicate（复制工具），快捷键为Shift+D，以及Spin（旋转挤压工具）和Screw（螺旋工具）。

前面已经简单介绍了挤压、旋转和缩放工具。其中挤压工具可以将边挤压成面，再将面挤压成体，挤压也因此成为建模中使用最频繁的工具之一。配合选取和移动G功能，在对多边形完成挤压操作后，可以使新建的多边形随鼠标做自由移动。

环切工具Ctrl+R能对物体做自定义切割，并在原物体基础上细分出更多的多边形。旋转R和缩放S没有创建多边形的作用，但是常被用于对多边形做形态上的调整。

### 3.3.2 平滑工具

多边形是网格物体的基础，但是在默认状态下多边形会保持平整的状态，边缘没有平滑的过渡，如图3-61左图所示，这样的模型在渲染后，效果将如图3-61右图所示。为了让模型在渲染时看上去更光滑，需要对其做一些平滑处理。

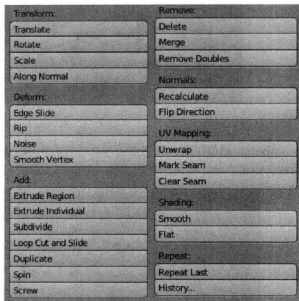


图 3-60 快捷工具栏菜单

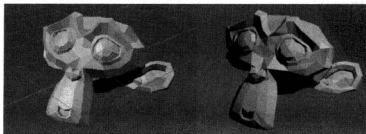


图 3-61 未平滑的多边形和渲染结果

首先在物体的编辑模式下，全选所有的面，单击 W 后就能看到两个平滑选项，Shade Smooth（着色器平滑）和 Smooth（平滑）。如图 3-62 上图所示，为选项菜单。前者能在保留模型细节的基础上，利用着色器计算来平滑多边形的表面，使其看上去完全光滑，而后者是直接平滑掉多边形的棱角，在物理层面上降低两个面之间的夹角角度，如图 3-62 下图所示，分别为两种平滑后的渲染效果。

可以看出，Shade Smooth 后的物体并没有丢失模型细节，因为着色器计算并不会改变物体本身的物理属性，而是从视觉上对物体做了平滑处理。但是物体在 Smooth 操作后，整个模型的尖锐轮廓都被打磨掉了，这时系统直接在网格结构上做出了修改，减小了面之间衔接边的夹角角度，但是面与面之间的不平整尖锐过渡依然存在。

Blender 也提供了 Auto Smooth（自动平滑）的功能，在编辑模式下进入 Object Data（物体数据）面板，找到 Normals（法向面板），勾选 Auto Smooth（自动平滑）选项后，调节下面的角度数字即可，如图 3-63 所示。开启自动平滑功能的物体不会修改网格本身的显示，而是会在渲染中自动地对夹角小于所填度数的棱角做平滑处理，这里的角度数字越大越能得到一个良好的平滑渲染效果。

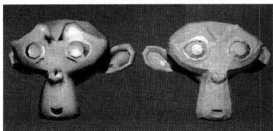
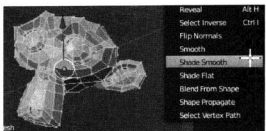


图 3-62 两种平滑的对比效果

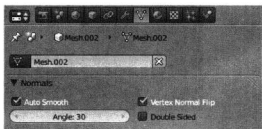


图 3-63 Auto Smooth（自动平滑）选项

### 3.3.3 使用关联复制实现镜像建模

在制作一些角色或特殊模型时，需要使用同步的对称式建模。前面曾介绍过关联复制，接下来就简单讲解如何使用关联复制来实现对称化建模。首先在物体模式中单击 Shift + C 定位光标至原点，然后单击 Shift + A 添加一个 Plane 物体，单击 Tab 键进入其编辑模式。全选网格后向 X 轴正向移动一定距离，然后退出至物体模式，如图 3-64 左图所示。接着单击 Alt + D 对物体做关联复制，不做任何改变时直接松开鼠标，单击 N 键进入属性菜单，在 Scale 中将 X 改为 -1，如图 3-64 右图所示，被关联复制的物体将以原点为轴，被镜像移动至 X 轴的负方向。

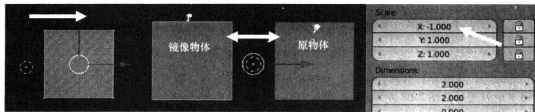


图 3-64 关联复制后镜像物体位置

再次单击 Tab 键进入原物体的编辑模式，由于镜像物体的关联属性，我们对原物体的任意编辑都将同步至新建的镜像模型上，效果如图 3-65 所示。

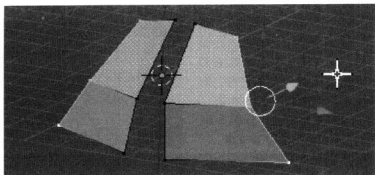


图 3-65 对称建模

### 3.3.4 螺旋和细分工具

建模中还有两个较常用的工具——Screw（螺旋）和 Subdivide（细分）工具。Screw 工具常用于制作螺旋形的物体，Subdivide 则能将所有物体一分为二。下面先了解一下 Screw 的用法。

在物体模式中添加一个 Circle 物体，按 Tab 进入编辑模式，如图 3-66 左图所示。接着单击 R + X + 90，Circle 则在 X 轴方向上旋转了 90 度，如图 3-66 右图所示。



图 3-66 使用数字键输入来旋转物体

在正视图视角中，新添加一个线段，并将光标定位在如图 3-67 左图所示位置。接着单击 Add 菜单中的 Screw，Blender 将以光标为原心，以线段方向为轴生成一个螺旋体，调节 Steps 和 Turns 等参数，即可得到如图 3-67 右图所示效果。

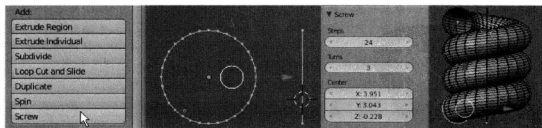


图 3-67 Screw（螺旋）工具

Subdivide（细分）有两种模式，一种是标准的细分模式，而另外一种是在细分的同时对物体做平滑处理。如图 3-68 左图所示，针对于一个标准的 Cube 物体，使用 Subdivide（细分）一次后，物体表面将被划分为  $2 \times 2$  的网格，二次细分后则被划为  $4 \times 4$  的网格，效果如图 3-68 中图所示。如果对同样的物体使用 Subdivide Smooth 2 次，物体不仅会被细分，同时还将会平滑掉所有的棱角边，效果如图 3-68 右图所示。



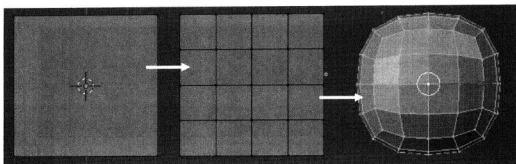


图 3-68 Subdivide (细分) 效果

### 3.3.5 精确切割工具

Knife Cut (精确切割) 工具能在建模中对多边形表面做自定义的划线切割, 快捷键为 K。如图 3-69 所示, 在一个标准 Plane 上, 按住 K + LMB 时鼠标会变成一个小刀形状, 在平面上滑动鼠标即可切出任意切割线, 完成后松开 LMB 确定, 系统会自动根据切割点对多边形进行细分重建。

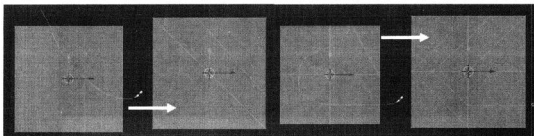


图 3-69 Knife Cut (精确切割) 工具

精确切割的前提是切割面必须处于被选择状态。在 T 工具菜单中可以找到属性选项。切割工具提供了三种切割方式: Midpoints 将保留切割点在边的中心位置, Multicut 允许在切割线时, 使用键盘数字键输入切割线的数量, Exact 则可精确地划分出切割点的位置。如图 3-70 所示分别为三种切割方式的效果。

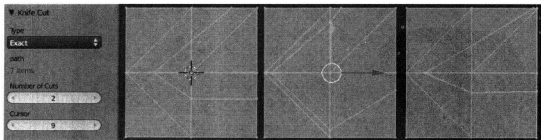


图 3-70 三种切割方式

### 3.3.6 基本建模工具的配合使用范例

下面简单地通过一个小例子来讲解挤压等基础建模工具的应用。

首先在物体模式中单击 Shift + A, 添加一个 Cube (正方形), 如图 3-71 左图所示。然后单击

Tab 进入其编辑模式，按 A 全选所有点，使用 S 缩小正方体，如图 3-71 右图所示。

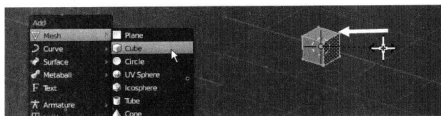


图 3-71 添加基础物体

使用组合键 Ctrl + Tab 调出选择模式菜单，切换至 Face（面）选择模式，如图 3-72 左图所示。然后使用 RMB 选择顶上的面，按 G + Z 向 Z 轴正上方移动一定的距离，如图 3-72 右图所示。

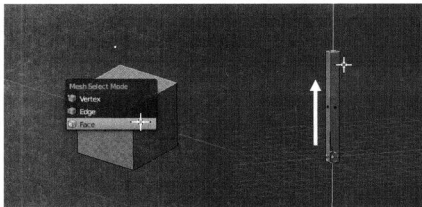


图 3-72 拉伸正方体

松开鼠标后，按 Ctrl + Tab 切换回点模式，单击 Ctrl + R 在立方体水平面上拉出一个环切线，单击一次 LMB 后，滑动鼠标向 Z 轴上方移动到如图 3-73 左图所示位置。接着重新回到面模式，使用 Shift + RMB 选择侧面的两个面，单击 E 后松开鼠标，然后再单击 S + Y 向 Y 轴方向做水平缩放操作，效果如图 3-73 右图所示。

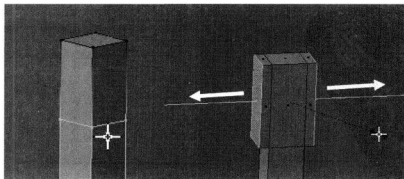


图 3-73 继续缩放

使用 Shift + RMB 选择 Y 负轴一侧的 3 个平面，按 E + G + Y 向负轴方向挤压一定的距离，如图 3-74 左图所示。同理，挤压 Y 轴正方向的 3 个面，松开鼠标后再单击 R + X，将平面在 X 轴上做一定的旋转，如图 3-74 右图所示。

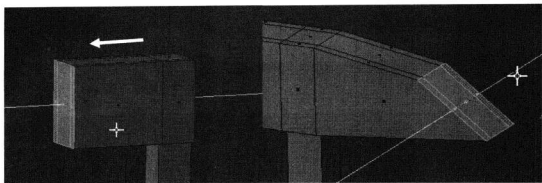


图 3-74 挤压和旋转

单击  $\text{Ctrl} + \text{Tab}$  进入点选择模式，选择顶上的点，按  $G + Z$  做一些水平调整，如图 3-75 左图所示。最后按  $\text{Tab}$  退回到物体模式，一个小榔头就做好了，效果如图 3-75 右图所示。

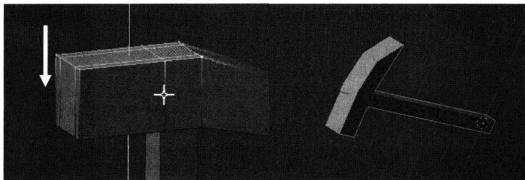


图 3-75 小榔头成品

接下来再次进入编辑模式，使用小键盘的  $\text{Num7}$  转换到顶视图视角，然后单击  $\text{LMB}$  将光标定位到如图 3-76 所示位置，按  $A$  全选小榔头，在工具菜单中单击  $\text{Spin}$ ，榔头将围绕光标做旋转复制。

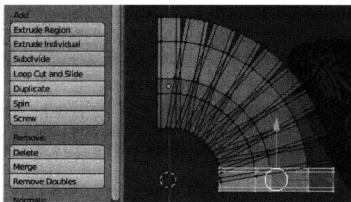


图 3-76 旋转复制

这时系统自动生成了 9 个小榔头，并且以光标为原点旋转了 90 度。在旁边的属性面板中修改这一默认设置，将  $\text{Steps}$ （步进）修改为 11， $\text{Degrees}$ （角度）设置为 360，这样 11 个小榔头则自动围着光标排列成一个圆圈，效果如图 3-77 所示。

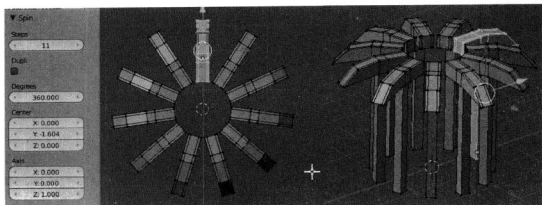


图 3-77 自定义 Spin 效果

## 3.4 高级网格建模

### 3.4.1 顶点组

Vertex Group（顶点组）是在一个多边形物体中，将网格上的一部分选择点编辑成特定的组，用于定义属性或功能在当前物体上的作用影响范围。例如在梳理毛发、绘制软体和设置多材质时，常会使用顶点组来划分出物体上不同的作用区域。如图 3-78 所示，对人头添加粒子毛发制作胡子时，希望粒子效果范围只影响下巴的区域，那么可以将这部分的点选中并编辑为一个 Jaw 顶点组，接着在粒子毛发面板中将粒子作用范围指向这个顶点组即可。

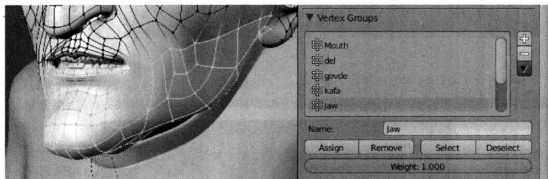


图 3-78 Vertex Group（顶点组）

顶点组又像书签一样，能帮助你在物体上快速标记出一部分点集合。复选择一部分的点或面后，在 Vertex 面板中单击十字星即可添加一个新的顶点组，使用 Assign（指派）功能也可将这些网格分配到已有的顶点组。面板中的 Select（选取）功能，能快速高亮选择模型中被标注的顶点组集合，反之 Deselect 能取消选择。

在编辑模式中也常使用顶点组来辅助建模，例如在制作一个乐高模型时，可以将乐高零件分别用顶点组作命名归类，然后通过选取顶点组的方式来选择物体，并通过复制操作来快速地制作模型。

顶点组的另外一个功能就是它在蒙皮和骨骼中的应用，如果需要指定一块骨骼去驱动一定范围内的网格，那么可以将这部分网格定义到一个顶点组，然后将其指向到对应的驱动骨骼即可，这些会在第 8 章详细讲解。

### 3.4.2 权重绘制

权重绘制常与顶点组配合起来使用，用于控制顶点组的影响衰减范围。

Weight paint（权重绘制）以渐变的色彩来标识顶点组对属性值的影响强度，常用于平滑顶点组之间的过渡。不同的颜色区分了不同等级的深度值，深度阶度图如图 3-79 所示，其中蓝色代表权重值为 0，也就是说当前网格将不受任何外界属性影响。越接近红色的权重值越大，最右边的红色权重值为 1，表示网格将完全接受属性参数对其的影响。如果在顶点组之间叠加地绘制上权重值，则可以为顶点组间的影响效果提供一个缓冲的过渡效果。




图 3-79 权重值的色彩阶度图

Weight Paint（权重绘制）模式可以从物体模式中单击 Ctrl + Tab 快速进入，进入界面后，网格将显示为权重的色彩深度图。单击 T 调出权重工具面板，如图 3-80 所示，就可以使用笔刷直接在物体表面上为当前顶点组绘制权重值了，例如在顶点组之间的交界处绘制出平滑的过渡权重值。



图 3-80 Weight Paint（权重绘制）模式编辑窗口

笔刷面板中, Weight (权重) 表示了当前绘制的权重值, 可以通过滑动滑块来改变值大小。当 Auto Normalize 选项被勾选后, 绘制区域内对应骨骼的顶点组将被自动设置影响深度为 1。Size (尺寸) 参数可修改笔触大小, 调节快捷键为 F。Strength (力度) 参数控制绘制的笔触力度, 调节快捷键为 Shift + F, 单击后通过滑动 MMB 来改变大小。如果激活旁边的  符号, Blender 将使用手绘板的笔尖压力感应来自动调节绘制力度。

### 3.4.3 雕刻模式

Sculpt Mode (雕刻模式) 是一种特殊的建模方式, 它允许在模型上使用类似雕塑的操作, 通过笔刷来模拟刻刀, 调整模型的形体并增加更多细节。要对物体执行雕刻就需要进入雕刻模式, 同权重模式一样, 可单击 Ctrl + Tab 快速进入, 也可在模式菜单中选择。如图 3-81 所示, 为雕刻模式的界面。



图 3-81 Sculpt Mode (雕刻模式) 编辑窗口

雕刻前需要对模型做多级细分, 使用足够的网格数才能表现出雕刻后的细节。但是系统会由于面数的增加而运行缓慢, 为了优化雕刻环境, 可以在操作过程中, 使用快捷键 Alt + B 来做框选隐藏, 如图 3-82 左图所示。如果要恢复显示, 只需要使用快捷键 Alt + B 即可。

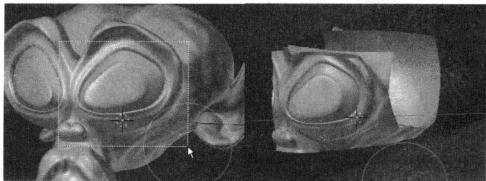


图 3-82 隐藏部分网格优化雕刻性能

### 3.4.3.1 Brush (笔刷) 菜单

雕刻模式下有几种常用笔刷, 如图 3-83 所示, 为笔刷菜单。

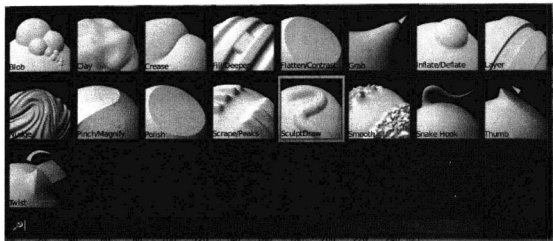


图 3-83 Brush (笔刷) 菜单

**Blob:** 球体笔刷, 它可以在物体表面添加球状凸起, 如图 3-84 所示, 为 Blob 笔刷的作用效果。

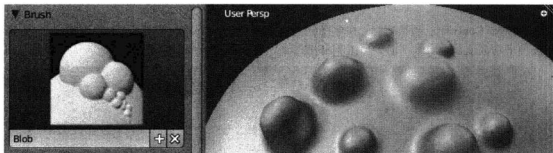


图 3-84 Blob (球体) 笔刷

**Clay:** 黏土笔刷, 能制造一种敷上泥土的感觉, 用于增加表面的凸起结构, 如图 3-85 所示。

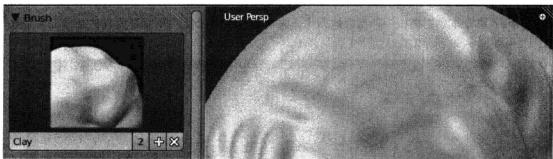


图 3-85 Clay (黏土) 笔刷

**Crease:** 折痕笔刷, 用于制作清晰的折痕边缘效果, 如图 3-86 所示。

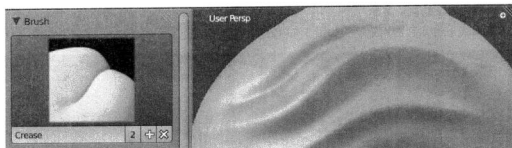


图 3-86 Crease (折痕) 笔刷

**Fill/Deepen:** 填充笔刷, 可以将凹槽的部分填充起来, 填充水平面根据物体的基础面而定, 如图 3-87 所示, 为 Fill 填充后的效果。

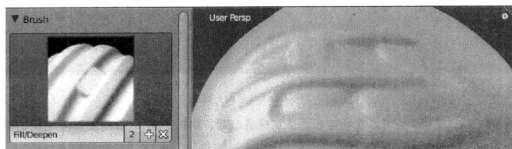


图 3-87 Fill/Deepen (填充) 笔刷

**Flatten/Contrast:** 展平笔刷, 按照凹凸的高低计算出一个平均高度面, 然后将这部分高低面按照平均高度整理成水平面, 如图 3-88 所示, 为展平一部分高低面后的效果。

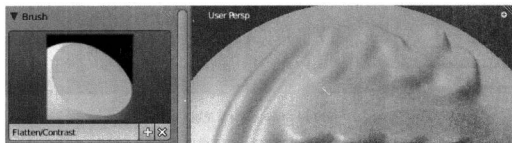


图 3-88 Flatten/Contrast (展平) 笔刷

**Grab:** 移动笔刷, 模拟一种抓取的动作, 来改变模型的结构, 和编辑模式中的移动功能类似, 效果如图 3-89 所示。

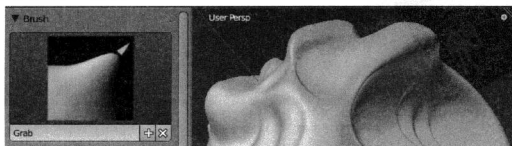


图 3-89 Grab (移动) 笔刷



**Inflate/Deflate:** 膨胀笔刷, 可以使被选中的顶点膨胀起来。不同于 Draw 笔刷使顶点表面直接突起, Inflate 的效果更类似于将整个顶点按法向方向突起, 效果如图 3-90 所示。

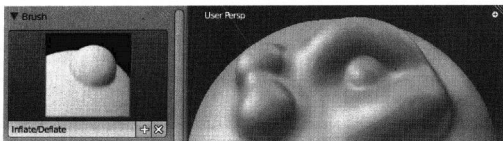


图 3-90 Inflate/Deflate (膨胀) 笔刷

**Layer:** 层笔刷, 可以按照固定的力度抬高或者降低模型的表面, 这个值由笔刷力度决定, 如图 3-91 所示, 为 Layer 笔刷的效果。

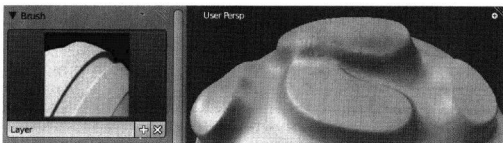


图 3-91 Layer (层) 笔刷

**Nudge:** 扭转笔刷, 用于制作一种旋转的扭曲效果, 如图 3-92 所示。

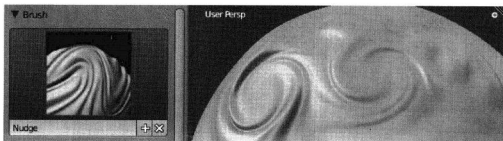


图 3-92 Nudge (扭转) 笔刷

**Pinch/Magnify:** 捏挤笔刷, 可以将顶点周围的面向内捏挤, 制作出坚硬的边缘效果, 如图 3-93 所示。

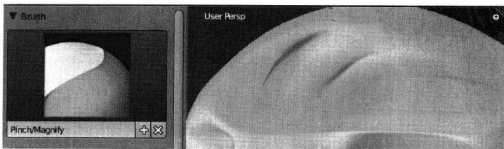


图 3-93 Pinch/Magnify (捏挤) 笔刷

**Polish:** 抛光笔刷，将表面打磨平整，如图 3-94 所示，为将一部分凹凸面打磨后的效果。

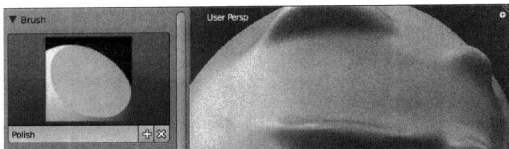


图 3-94 Polish（抛光）笔刷

**Scrape/Peaks:** 削刮笔刷，可以将凸起的模型削掉，如图 3-95 所示，为刮去一部分凸起面的效果。

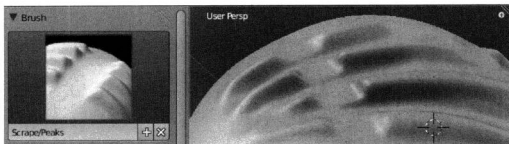


图 3-95 Scrape/Peaks（削刮）笔刷

**SculptDraw:** 刻画笔刷，可以在表面上绘制一段突起效果，目的是让顶点向外凸起，如果按住 Alt 可以绘制向下凹的效果，如图 3-96 所示。

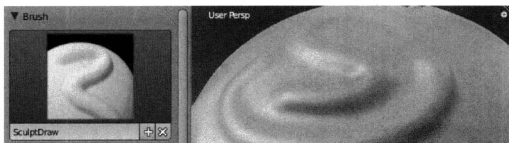


图 3-96 SculptDraw（刻画）笔刷

**Smooth:** 平滑笔刷，可将模型表面的尖锐或不平整部分光滑掉，如图 3-97 所示，为平滑后的效果。



图 3-97 Smooth（平滑）笔刷

**Snake Hook:** 蛇钩笔刷, 用于在表面拉出渐细的挂钩效果, 如图 3-98 所示。

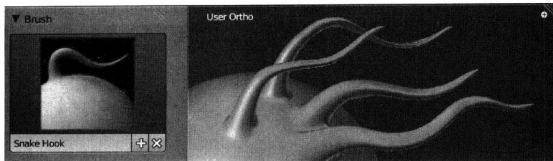


图 3-98 Snake Hook (蛇钩) 笔刷

**Thumb:** 指推笔刷, 模拟一种用手指在表面推动黏土的效果, 如图 3-99 所示。

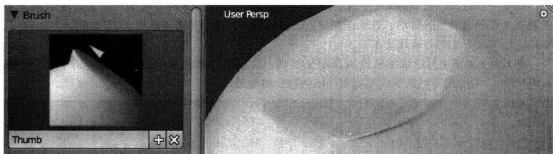


图 3-99 Thumb (指推) 笔刷

**Twist:** 扭曲笔刷, 制作出扭曲的翻转效果, 如图 3-100 所示。

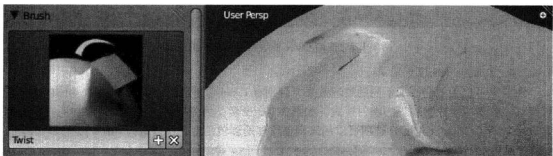


图 3-100 Twist (扭曲) 笔刷

**Size:** 笔刷尺寸, 用于控制笔刷的影响范围, 快捷键为 F, 单击后只需要滑动 MMB 即可灵活地改变画笔大小, 也可使用键盘输入数字来精确控制画笔大小。

**Strength:** 笔刷强度, 控制着笔刷在模型上的动作力度, 较高的数值可以为模型添加较大幅度的变化, 也可以使用组合快捷键 Shift + F 来改变笔刷的强度。

**Add/Sub:** 凸起和凹陷效果, 用于控制笔刷的影响模式。

**Accumulate:** 堆积模式, 激活后允许点在凸起位置产生堆积的效果。

#### 3.4.3.2 Stroke (绘制) 菜单

**Airbrush:** 喷枪, 当处于激活状态时, 单击笔刷会持续添加雕刻效果, 直到松开鼠标, 快捷键

为 Shift + A。当关闭时，笔刷对物体的改变将没有持续效果，只有 Grab 笔刷不能使用喷枪功能。

**Anchored:** 表面映射，允许在表面上拖拽出纹理的形状，效果如图 3-101 所示，快捷键为 A。

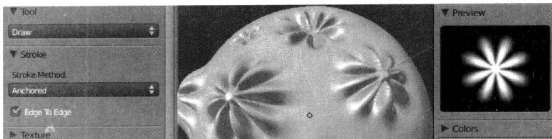


图 3-101 Anchored（表面映射）效果

**Space:** 间隔，用于控制笔触间的间隔距离，值为 1 时笔刷绘制的是连续不间断的效果。

**Dots:** 开启 Smooth Stroke（平滑绘制），用于实现一种自动平滑的效果。其中斜度选项定义了笔刷是否跟随画笔的移动，自动调整材质图案的角度，这个功能不适用于 3D 纹理。

#### 3.4.3.3 Curve 笔刷的曲线菜单

在 Curve 菜单中可以添加和使用自定义笔刷，这里使用曲线来描述不同的笔刷形状，如图 3-102 所示，为使用不同的曲线笔刷制作出的特殊雕刻效果。

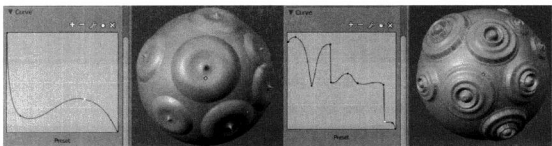


图 3-102 使用 Curve 曲线自定义笔刷的形状

#### 3.4.3.4 Texture（纹理）菜单

在这里可以为笔刷添加不同的纹理效果，分别有 Fixed（固定）、Tiled（平铺）和 3D 效果三种模式，分别用于控制纹理的映射模式，如图 3-103 所示，为分别使用三种模式添加的自定义笔刷效果。其中 Angle（角度）选项可以用来调整纹理的旋转角度，快捷键为 Ctrl + F。

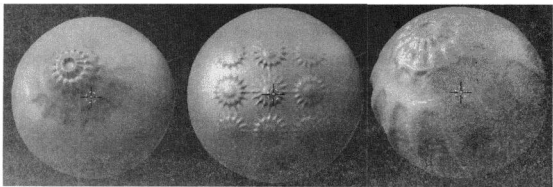


图 3-103 笔刷的纹理映射方式

### 3.4.3.5 Option (选项) 菜单

**Unified Setting:** 统一所有笔刷的尺寸和强度值。

**Lock:** 轴锁定选项, 可以屏蔽在对应坐标轴上做的改变。

**Show Brush:** 显示笔刷, 使笔刷大小在绘制过程中可见。

**Fast Navigate:** 快速导航, 当移动或者转动网格时, 系统将自动转换网格为低分辨率预览, 用于提升视图转换时的显示性能, 效果如图 3-104 右图所示。

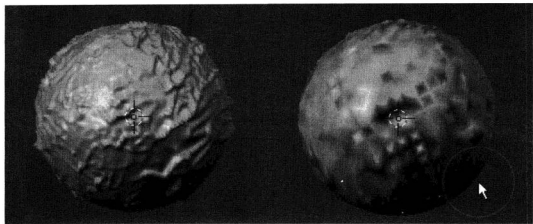


图 3-104 Fast Navigate (快速导航) 模式

**Symmetry:** 镜像选项, 当选择对应的坐标轴 X、Y 或 Z 后, 对模型所作的雕刻变化将对应地出现在镜像坐标上。

## 3.5 曲线建模

曲线和曲面也是一种 Object (物体), 区别于多边形等网格物体, 它们在空间中的表达方式略有不同, 曲线是一定数量的矢量线段组合, 而网格更像是一系列的点的集合。Blender 中集成了 Bezier (贝济埃) 曲线、NURBS (Non-Uniform Rational B-Spline, 非统一有理 B 样条) 曲线和 Surface (曲面)。

和网格使用“控制杆”的编辑模式不同, 曲线和曲面使用的是“控制点”来修改线条。控制点采用了数学有理表达式的控制方式, 它具有精确表现二次曲线和二次曲面的能力, 可以制作出一些复杂的曲面造型和特殊效果。但是并不是所有的造型都适合由曲线来表现, 一些有明显棱角边的物体并不适合使用曲线来制作。

由于曲线使用了较少的数据量来控制物体结构, 因此在建模的时候它们可以在消耗较少内存的情况下, 表现出极好的视图性能和预览效果。但是在曲线的渲染阶段, 系统会对曲线模型做一定的数据转换, 所以渲染的时间会稍有增加。

### 3.5.1 曲线的编辑

和网格的编辑方式类似, 扩展曲线上的控制点即可延长曲线长度, 使用快捷键 E 并结合 G 可在选择点上挤压出新的节点, 如图 3-105 所示。但是, 如果同时选择多个点, 挤压功能将会失效, 所以对曲线的扩展只能在单个点上执行。

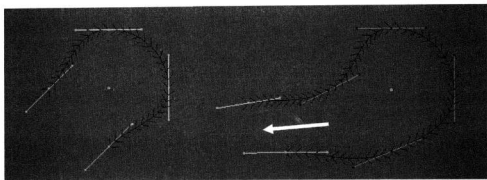


图 3-105 曲线的控制与扩展

如果需要封闭一个曲线段时，只需要单击快捷键 **Alt + C**，系统就会自动缝合曲线的两端，如图 3-106 左图所示。如果是 2D 的曲线，可以单击 **F** 将线段连接起来，曲线将自动建立起封闭面，并且在渲染中可见，如图 3-106 右图所示。

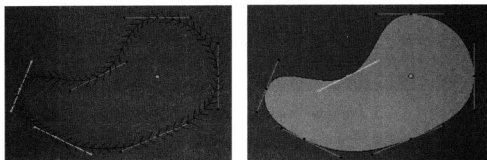


图 3-106 Cyclic（封闭曲线）效果

### 3.5.1.1 Shape（外形）面板

如图 3-107 所示，为 Shape（外形）的控制面板。

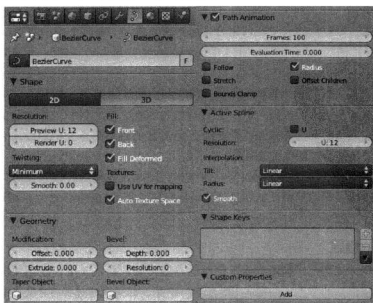


图 3-107 Shape（外形）控制面板

**2D/3D:** 2D 选项状态下的曲线只允许在平面上执行编辑操作, 系统将自动锁定其垂直方向上的坐标控制杆。开启 3D 选项后, 曲线才可在三维空间中做自由变换。

**Resolution:** 解析度, 线段的解析度决定了线段的疏密度。曲线可以被看作是一条连续的线段, 因此解析度越高的曲线线段看上去越平滑。当解析度较低时, 曲线将失去弧线的形态, 变得趋于线性化。如图 3-108 所示, 从左至右分别是解析度为 1、3 和 9 的曲线效果。



图 3-108 不同解析度的曲线平滑度

**Twisting:** 扭转, 用于设置曲线扭转时, 样条法线的方向。

**Fill:** 用于封闭曲线扩展后的多边形, 其中 Front 将封闭正面, Back 可封闭背面, Fill Deformed 则封闭应用 Shape (形变) 和修改器后的曲线。

**Textures:** 自动调整曲线物体在赋予纹理后的坐标空间。

### 3.5.1.2 Geometry (几何结构) 面板

**Modification:** 修改曲线的立体结构, 其中 Width (广度) 用于控制曲线在原心范围内的缩放效果, Extrude (挤压) 参数能将曲线扩展成平面, 如图 3-109 所示。

**Bevel:** 倒角, 能为挤压出平面的曲线物体添加倒角效果, 如图 3-109 所示, 分别为其添加一定的挤压和倒角效果, 其中 Resolution (解析度) 参数值越大, 倒角越平滑。

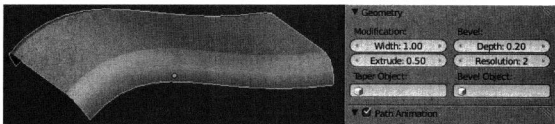


图 3-109 Extrude (挤压) 与 Bevel (倒角) 效果

**Taper/Bevel Object:** 锥化/倒角模式, Taper 可以将当前曲线圈围绕另一个曲线轴放样形成一个锥化物体, Bevel 则可以将倒角模式应用到当前的曲线物体上。例如, 添加一个圆作为曲线物体的直径形状体, 再添加一条曲线来控制物体的锥化形态。这时就可以分别通过对 base 和 taper 的修改控制, 来调整当前曲线物体的形状, 效果如图 3-110 所示。



图 3-110 Taper/Bevel (锥化) 与 (倒角) 模式

### 3.5.1.3 Path Animation (路径动画) 面板

**Frames:** 定义了路径动画的帧范围。

**Follow:** 启用跟随动画效果, 适用于物体使用曲线为运动路径的动画。

**Stretch:** 启用拉伸效果, 适用于物体的变形动画。

**Radius:** 使用曲线的法线半径来影响路径的跟随动画效果。

**Offset Children:** 使子物体与路径保持一定的偏移量。

#### 3.5.1.4 Active Spline (活动样条曲线) 面板

**Cyclic:** 将曲线的首尾在 U 坐标上自动连接, 封闭起来。

**Resolution:** 表面解析度, 该值越高, 对应曲线的扩展面越平滑。如图 3-111 左图所示, 为 U 值为 5 的效果, 而右图为 U 值增大至 30 的效果, 可以看到网格的细分数明显增多。

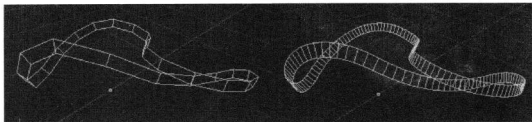


图 3-111 曲线的 Resolution (表面解析度)

**Interpolation:** 定义了曲线的插补属性, 可分别选择 Tilt (倾角) 和 Radius (半径) 两种控制模式。

**Smooth:** 用于平滑曲线的放样表面。

### 3.5.2 Bezier 曲线

Bezier 贝济埃曲线由线段和控制节点组成, 其中节点是可拖动的控制支点, 用于调整曲线的形状变化。Bezier 曲线不仅可用于建模, 也常在动画模块中使用, 提供精确的控制路径等数据。在 3D 窗口中新添加一个 Bezier 曲线, 单击 Tab 进入编辑模式。如图 3-112 所示, 为一个初始化状态的 Bezier 曲线线段, 左边的工具栏显示了一些常用的曲线工具。曲线中的点 H 就是控制杆, C 是曲线的控制节点, 节点间的 B 就是曲线线段, 线段上的箭头方向代表曲线的法线方向和线条密度。

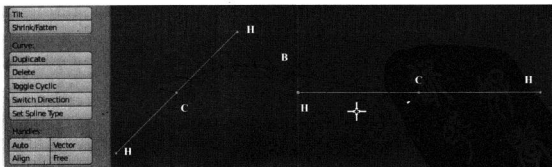


图 3-112 Bezier 曲线

对控制杆和节点的选择方式, 和网格中对点的选择方式一样, 单击 RMB 选择后即可对其做相应的形变操作, 例如使用快捷键 G 做移动, 使用快捷键 S 做缩放, 使用快捷键 R 做旋转。这里控制杆有 4 种控制模式。

**Auto Handle:** 自动模式, 控制杆能根据曲线当前节点自动调整长度和方向, 辅助制作出平滑的曲线效果。控制杆此时呈黄色, 快捷键为 Shift + H, 效果如图 3-113 左上图所示。

**Vector Handle:** 矢量模式, 在这种模式下曲线将被调整为矢量的线段形态, 控制杆变成绿



色, 快捷键为 V, 效果如图 3-113 右上图所示。

**Align Handle:** 连续模式, 这时控制杆将始终处于节点的切线方向位置, 曲线呈现连贯的弧度, 控制杆呈红色, 快捷键为 Shift + H, 效果如图 3-113 左下图所示。

**Free Handle:** 自由模式, 控制杆在这种模式下互相独立, 可自由旋转, 可使用快捷键 H 来转换, 控制杆呈黑色, 效果如图 3-113 右下图所示。

也可以使用工具菜单中的 Handles (控制杆) 选项, 来修改当前控制杆模式, 如图 3-113 左侧所示。

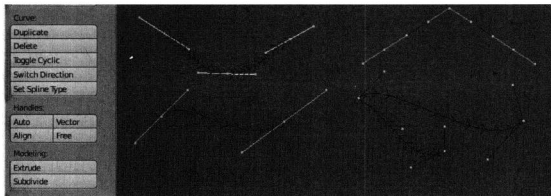


图 3-113 Handle (控制杆) 模式

### 3.5.3 NURBS 曲线

NURBS 曲线发展于 20 世纪 50 年代, 它由那些希望在车体或船壳表面上使用精确的自由曲面数学表达式的工程师们所发明, 它最大的特点就是曲线数据可以被任何技术精确地复制出来。现在, NURBS 已经被广泛应用于计算机辅助设计, 并成为了工业标准的一部分。

Blender 中的 NURBS 曲线是由带比重控制的向量节点集合组成, 与 Bezier 曲线的最大区别就是, 它的控制方式采用了权重值来定义控制点和曲线顶点之间的距离, 因此更趋近于一条完美的曲线。NURBS 上的节点决定了对应曲线段上点分布的局部密度, 所以调节 NURBS 曲线的控制点权重就可以直接改善一条曲线的形状。

组成一个 NURBS 曲线段的最少控制点数目等于一条曲线的 Order (阶数) 加 1, 再加上控制点的数目。其中曲线的 Order (阶) 涉及产生曲线的数学公式中的高次幂指数, 因此, 曲线的 Order (阶) 越高, 产生该曲线所需要的计算量也就越大。一阶曲线相对应于一条直线段, 二阶曲线相对应于一条二次曲线, 而三阶曲线则相对应于三次曲线。曲线的阶数越高, 曲线段所需要的控制点也就越多。

在创建 NURBS 曲线之前, 一定要把曲线的阶数和跨度等原理掌握清楚, 它们是产生一条“完美”曲线的决定性因素。如图 3-114 左图所示, 在 Blender 中添加一个新的 NURBS 线段, 默认的曲线为 Uniform (均衡控制), 这时的曲线在开放的末端不容易定位, 解决的方法是激活 Endpoint (端点) 控制, 这时曲线的两端将始终吸附在控制点的末端, 如图 3-114 右图所示。



图 3-114 曲线的 Endpoint (端点) 控制方式

Order (阶) 是 NURBS 曲线用于定义弧度的重要参数之一, 如图 3-115 所示, 左边为 2 阶曲线, 右边为 4 阶曲线的效果。Blender 提供了 2 到 6 阶数的选择范围, 低阶的曲线近似于线性化, 高阶的曲线弧度将相应地增加。



图 3-115 曲线的 Order (阶) 控制

Weight (权重值) 决定了曲线控制点的影响力, 节点的权重范围可以由 0.1 至最大 100, 越高的权重越能表现更强的控制约束。如图 3-116 所示, 分别对右下角的点设置权重值为 20 和 100, 可以看出权重影响了点与点之间的作用力, 拥有高权重的控制点可以产生更大的曲线密度。

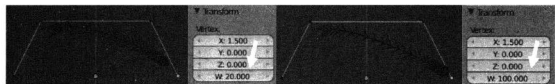


图 3-116 曲线的 Weight (权重) 控制

## 3.6 文字建模

Blender 的文字输入工具简单快捷, 只需要按住 Shift + A 键在弹出菜单中选择 Text 物体即可。单击 Tab 键进入文字的编辑模式后, 使用键盘的退格键消除默认的字符, 即可输入新的文字了。

### 3.6.1 文字的输入

如图 3-117 所示, 进入文字的编辑模式后, 标题栏菜单将出现 Text (文字) 菜单, 可以在 Special Characters (特殊符号) 菜单中快速找到一些常用的特殊符号。如果是大量的文字, 可以将文字保存在 txt 文本文件中, 单击 Paste File 打开文件即可自动导入已录好的段落文字。

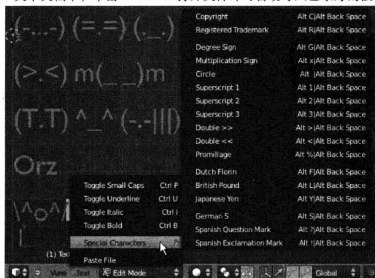


图 3-117 文字输入菜单

### 3.6.2 文字的编辑与修改

在文字编辑模式下，可以在编辑菜单中找到文字的编辑面板，如图 3-118 所示。

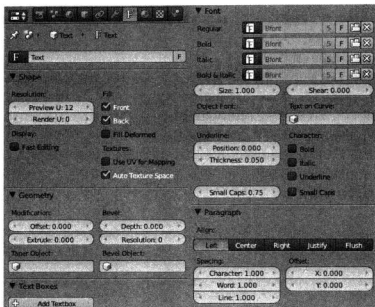


图 3-118 文字编辑面板

#### 3.6.2.1 Shape (外形) 菜单

**Resolution:** 解析度，数值越低，文字看上去越尖锐，反之越平滑。

**Display:** 开启 Fast Editing 显示功能，可以在编辑的时候取消文字的自动填充效果。

**Fill:** 文字的扩展显示，同曲线选项类似。

**Textures:** 纹理选项，用于自动调整文字物体上的纹理空间。

#### 3.6.2.2 Geometry (几何造型) 菜单

**Modification:** 修改曲线结构，Width (广度) 值用于控制文字的粗细缩放效果，Extrude (挤压) 值则能将文字扩展成立体。

**Bevel:** 倒角参数，能为挤压出立体效果的曲线物体添加边缘倒角，Depth (深度) 值控制倒角的宽度，而 Resolution (解析度) 越大，倒角越平滑。如图 3-119 所示，为添加了倒角和厚度的文字物体效果。

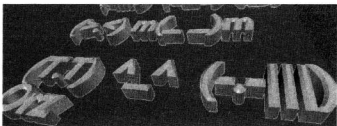


图 3-119 文字物体的几何造型

#### 3.6.2.3 Font (字体) 菜单

可以为输入的文字选择特定的字体文件，单击旁边的文件按钮，选择系统中的字体

“\*.font”文件即可。使用 Size 可调节文字的大小，使用 Shear 可设置文字倾斜的角度。

**Underline:** 添加下划线效果，其中 Position 用于控制下划线的位置，Thickness 调整下划线的粗细。

**Character:** 字体效果，这里标示了三种字体效果——Bold（加粗）、Italic（斜体）和 Underline（下划线）。

#### 3.6.2.4 Paragraph（段落）菜单

**Align:** 段落排列，这里有常用的 5 种排列方式，其中 Left 为左对齐，Center 为置中，Right 为右对齐，Justify 为两端对齐，Flush 为两端留边对齐。

**Spacing:** 间距，用于控制字符之间的间距距离。

**Offset:** 偏移，控制段落与对齐轴之间的偏移量。

至此，已经介绍了大部分常用的物体建模工具，除此之外，Blender 还提供了 Surfaces（曲面物体）和 Meta（变形球物体）等建模方式，但是在一般的工作项目中，这些物体并不常用，所以感兴趣的读者可以自行研究一下。

## 第4章

## 修 改 器

Modifiers（修改器）是物体的一种高级编辑模式，它提供了快速而高效的属性定义方法。对于一些重复而且繁琐的修改，例如镜像和细分，都可以直接为物体添加修改器来代替重复操作。不同于直接对物体作物理修改，修改器对物体属性的改动并不影响物体本身，而是仅修改物体的显示属性和渲染属性。如果需要完全修改物体的本体属性，只能应用修改器的效果，将修改器的更改烘焙至物体的物理属性中。如图4-1所示，为Modifiers（修改器）面板。

首先选择物体，然后单击 Add Modifier（添加修改器）按钮，在下拉菜单中单击选择需要添加的修改器即可。一个物体支持多修改器的叠加效果，执行叠加计算的顺序在面板中是自下向上的，因此同样的修改器将由于排列顺序不同，产生不同的物体修改效果。

如图4-2所示，为一个球体添加了两个修改器，左图为先执行 Mirror（镜像）运算，再做 Subsurf（细分）处理的修改效果，右图则为先做 Subsurf（细分），再执行 Mirror（镜像）修改的效果。可以看到，左图在低模的情况下先做镜像处理，由于中间接缝处的点数目较少，所以镜像的 Clipping 计算不会将点向内收紧，最后的效果比较平滑。而右图的接缝处由于细分后表面点数目的增加，镜像修改器在高模的基础上，使用 Clipping 计算把接缝边向内收紧，形成一个比较尖锐的对称过渡效果。

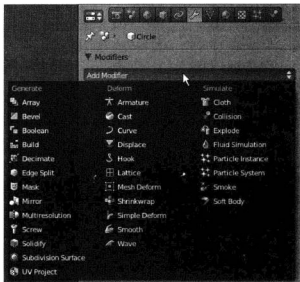


图4-1 Modifiers（修改器）面板

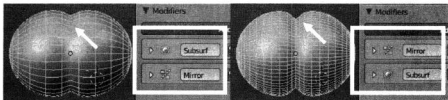


图4-2 修改器顺序对修改效果的影响

Blender 下的修改器大致分为三类，分别是：

- ❑ **Generate**：结构类修改器，这一类修改器对物理结构做出变形更改，但仅运用于物体的网格结构部分，例如镜像修改器和细分修改器。
- ❑ **Deform**：形变类修改器，这类修改器常用于改变物体的外部形态，包括曲线修改器、置换修改器和晶格修改器等。
- ❑ **Simulate**：模拟类修改器，这类修改器是一种数据读取和计算型修改器，它可以从物体上读取网格等数据，并运用在相应的特殊效果中，例如骨骼修改器和毛发修改器等。

## 4.1 结构类修改器

### 4.1.1 阵列修改器

物体 Array（阵列）是指系统将一个原始物体，按照具有一定规律的排列顺序，采用复制的方法创建出一组物体矩阵。其中复制方式采用了关联复制，也就是说对基物体的修改将关联影响生成的所有新物体。当需要创建一个具有大量重复规律的物体时，就可以采用这个修改器。如图 4-3 所示，为 Array（阵列）修改器的参数面板。

**Fit Type**：控制方式，这里提供了三种控制阵列物体的方式。Fit Curve（曲线匹配），它能将阵列物体的排列匹配到一条指定的曲线路径上。Fit Length（距离匹配），则将阵列物体匹配至一条固定 Blender 单位的直线路径上。前两种方式都是根据曲线和直线的距离，自动计算需要增加的新建阵列物体数目，而 Fixed Count（固定数量）和这两种模式相反，它能直接控制新增加的阵列物体数量。如图 4-4 所示，从上至下分别为三种不同控制方式的阵列效果。

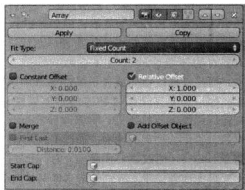


图 4-3 Array（阵列）修改器的参数面板

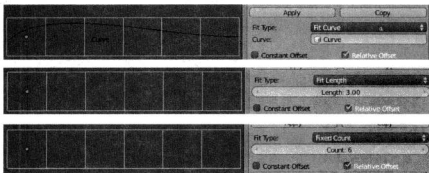


图 4-4 三种不同的控制方式

Fit Curve 和 Fit Length 使用了本地坐标系来控制阵列物体的缩放和新建数量的计算等操作，所以当在物体模式下缩放物体时，修改器会同比例地缩放所有新建阵列物体的尺寸，同时由于 Length 在本地坐标系中也产生了相应的缩放，所以物体模式下的操作不会改变新建阵列物体的数量。如图 4-5 上图所示，为缩放前的物体效果，当在物体模式下将物体缩小 50% 后，阵列的效果将如图 4-5 中图所示。如果在编辑模式中对物体执行缩放操作，修改器则会自适应地添加复制物体来满足控制长度的匹配，如图 4-5 下图所示，为编辑模式中对物体缩小 50% 后的效果。

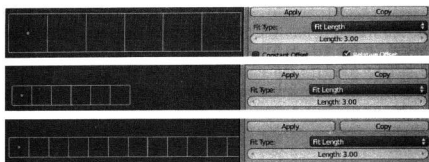


图 4-5 不同坐标系中缩放物体在阵列修改器下的效果

**Constant Offset:** 常量偏移，用于在新增加的物体之间添加一个固定的位移偏移量。如图4-6上图所示，每个新增加的物体都会在前一个物体的位置基础上，向 X 正轴偏移 1 个 Blender 单位常量。

**Relative Offset:** 相对偏移，不同于常量偏移，相对偏移的偏移量不是 Blender 单位，而是物体边界框的尺寸比例。如图4-6下图所示，为 Cube 的阵列物体设置偏移量为 1 时，新建阵列物体之间的间隔距离即为物体本身的尺寸。

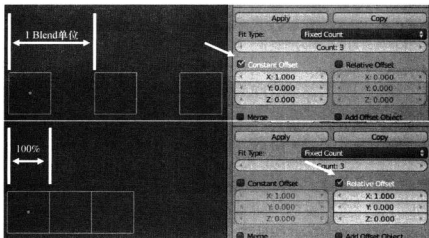


图 4-6 两种 Offset（偏移）设置的区别

**Add Offset Object:** 物体控制偏移，前面的两种偏移方式并不适合自定义阵列的控制，这里可以使用一些物体的坐标数据来控制阵列的偏移量。如图4-7所示，首先在物体原点处添加一个 Empty（空）物体，然后将其名称添加至阵列修改器中的物体控制选项，这样通过旋转和移动 Empty（空）物体的方式即可控制 Cube 物体的偏移和排列方式了。

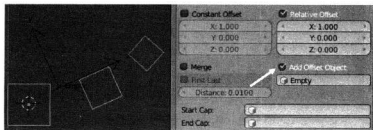


图 4-7 Object（物体）控制自定义偏移

**Merge:** 合并, 当激活时, 新建阵列物体间交接处的点将被自动缝合起来。其中 First Last (首尾) 选项, 用于开启物体首尾连接的功能。为阵列添加一个 1.05 个单位的 X 轴正向偏移, 如图 4-8 上图所示, 是未激活自动缝合功能时的阵列物体效果。激活合并选项, 设置合并范围为 0.05, 物体的接缝处表面将被自动合并了起来, 效果如图 4-8 下图所示。

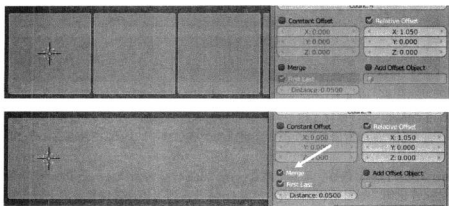


图 4-8 Merge (自动合并)

**Start/End Cap:** 添加阵列首尾的显示物体。在 Start Cap 首物体名称输入框中输入显示物体的名称后, 物体将被添加至阵列的首位置, 同理也可以添加一个物体作为阵列的尾显示, 效果如图 4-9 所示。

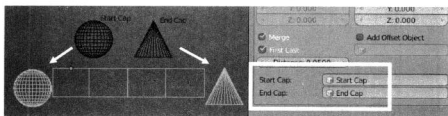


图 4-9 Start/End Cap (设置首尾物体)

接下来制作一个简单的阵列复制范例。

首先使用 Shift + C 复位光标, 单击 Shift + A 添加一个 Cube 物体, 然后再次单击 Shift + A 添加两个 Empty 物体, 分别命名为 ContV 和 ContH, 如图 4-10 所示。

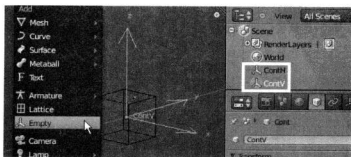


图 4-10 添加 Cube 和 Empty 物体

使用 RMB 单击选择 Cube, 在修改器面板中添加一个 Array (阵列) 修改器, 选择 Fixed



Count（固定数量）模式，设置固定复制次数为 20，如图 4-11 所示。同时勾选 Relative Offset，并取消所有的相对偏移量，单击 Add Offset Object，在下面的输入框中输入 ContV。

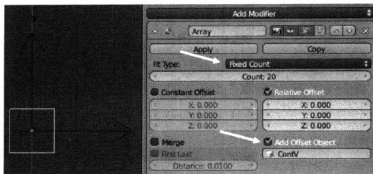


图 4-11 添加第一个 Array 修改器

接下来再添加一个阵列修改器，如图 4-12 所示，同样选择固定数目的复制模式，修改数字为 10。取消相对偏移量，单击 Add Offset Object，输入控制的物体名称 ContH。

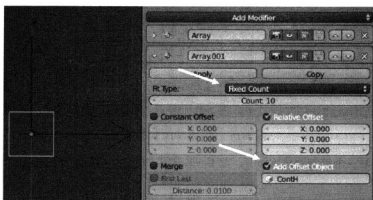


图 4-12 添加第二个 Array 修改器

由于两个 Empty 在 3D 环境中处于重叠状态，所以可以打开大纲管理器，在其中快速地选择不同的 Empty 物体，如图 4-13 左图所示。首先选择 ContV 控制，单击 S 缩小一定的比例，效果如图 4-13 右图所示。



图 4-13 在大纲管理器中可方便地选择重叠物体

接下来单击 R 对 ContV 稍微做一点旋转，新建的复制阵列效果如图 4-14 所示。

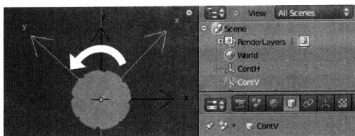


图 4-14 旋转 ContV 后的效果

按 G 将控制物体 ContV 向 X 轴正向移动一定的距离, 这时可以看到 Cube 群在阵列修改器的作用下发生了变化, 由中心向四周发散开来, 效果如图 4-15 所示。

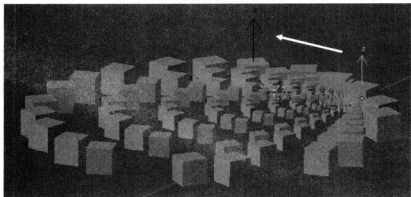


图 4-15 移动 ContV 后的效果

接下来选择另一个控制物体 ContH, 按 G + Z 向 Z 轴负方向移动一定的距离, 效果如图 4-16 所示。

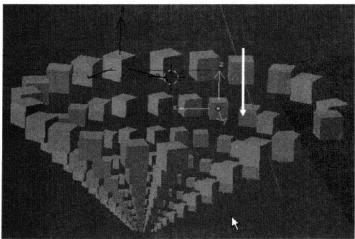


图 4-16 移动另一个控制器 ContH

再次单击 R + Z 将 ContH 在 Z 轴方向上做一定的转动, 使用鼠标 MMB 转换视图, 就可以得到一个十分漂亮的规律螺旋图了, 效果如图 4-17 所示。

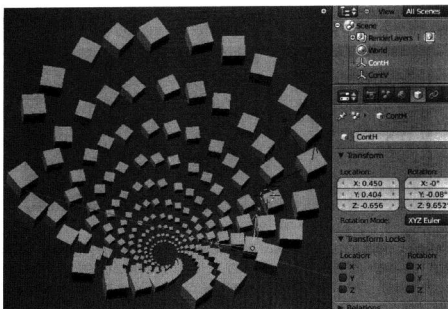


图 4-17 最后效果图

### 4.1.2 倒角修改器

常规的临界边都是由两个面组成的转角，倒角的概念就是使用一个斜面来平滑棱角边的边缘，用于降低两个面之间的相邻夹角，使物体的边缘在细分后显得更平滑。倒角修改器常用于建筑和家具的制作，如图 4-18 所示，左边是一个常规的正方体，右边则是使用默认参数修改的倒角效果。

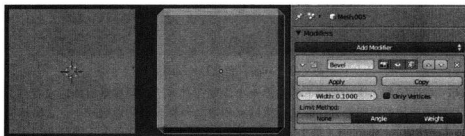


图 4-18 物体的 Bevel（倒角）效果

**Width:** 宽度是倒角修改器的重要参数，它定义了倒角的宽度，调节范围从 0 到无穷大。当 Width 为 0 时物体没有倒角效果，数字越大临界边上的倒角面越宽，如图 4-19 所示，从左至右分别是 0.1、0.2 和 0.5 的倒角效果。

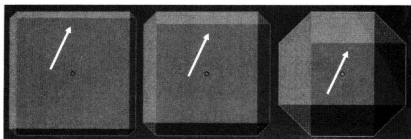


图 4-19 不同 Width 值的倒角效果

**Only Vertices:** 点模式，当激活这个功能后，倒角效果将从边上移除，仅显示在物体的点位置上，如图 4-20 所示，从左至右分别是 0.1、0.2 和 0.5 的点倒角效果。

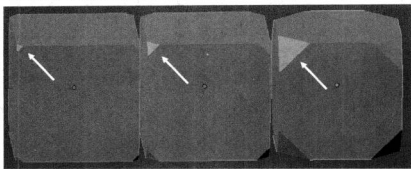


图 4-20 倒角只应用于点上的效果

**Limit Method:** 限制方式，在默认情况下物体的倒角不受任何附加外力的限制，但是当激活不同的限制方式（如 Angle（角度）和 Weight（权重））后，物体倒角会根据外力产生相应的形变。

**Angle:** 角度限制，当激活角度限制后，选项下面会显示一个角度数字选项，这里的数字规定，一旦两个边的夹角度数小于这个数值时，Bevel 将不会对这个边执行倒角编辑。如图 4-21 所示，设置限制角度为 60 度，那么大于 60 度的余角边将被执行 Bevel 效果，而小于 60 度的边则没有 Bevel 效果。



图 4-21 Angle（角度）控制倒角

**Weight:** 权重限制，这种选项下的物体至少需要拥有多个不同权重值的边或者点，高权重的点和边受到修改器的影响最大，而低权重的点和边受到修改器的影响较小。它的相关选项包括 Average（平均值）、Sharpest（最小值）和 Largest（最大值），这三个选项分别用于计算未标注权重点和边的网格倒角效果。如图 4-22 所示，左图为 Average（平均值）的倒角效果，可以看到其中箭头处未标注权重值的点，将按照权重的平均值 0.75 来计算倒角效果，中图则为采用权重 Sharpest（最小值）的限制效果，未标记的点将取权重值中的最小值 0.5，而右图为 Largest（最大值）的效果。

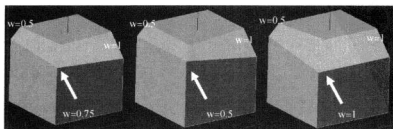


图 4-22 Weight（权重）控制倒角的效果

### 4.1.3 布尔运算修改器

Boolean（布尔运算）修改器可以对两个物体执行交集、并集和差集运算，从而得到一个新的混合物体。布尔修改器提供了3种布尔运算方式，分别为 Intersect（交集）、Union（并集）和 Difference（差集）。如图4-23所示，是 Boolean 修改器的参数面板。



图 4-23 Booleans（布尔运算）修改器的参数面板

在布尔运算中，两个原始物体被称为运算对象，其中一个为运算对象 A，另一个为运算对象 B。在建立布尔运算前，首先要在视图中选择运算对象 A，然后单击添加 Boolean 修改器，在 Object 输入框中输入运算对象 B 的名称，选择对应的模式即可进行物体间的布尔运算。

**Intersect：**交集模式，用来将两个物体的相交部分保留下来，并自动删除不相交的部分。如图 4-24 左图所示，将 Mesh 称为对象 A，将 Mesh.01 称为对象 B。首先单击 RMB 选择对象 A，为其添加一个 Boolean 修改器，在 Object 中输入运算对象 B 的名称，它们的交集网格就被自动运算出来了，如图 4-24 右图中箭头所示。

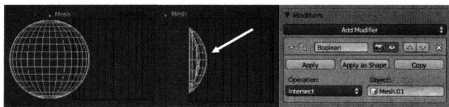


图 4-24 Intersect（交集）运算

**Union：**并集模式，用来将多个物体做合并计算，运算完成后的两个物体将合并成为一个新物体，并自动删除相交的部分。例如利用刚才新建的两个物体，修改对象 A 的修改器 Operation 选项，将运算模式修改为 Union，效果如图 4-25 所示，可以看到两个物体在相交的位置实现了自动缝合，并且相交部分的物体被自动排除并显示为黑色。

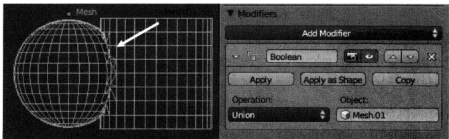


图 4-25 Union（并集）运算

**Difference：**差集模式，将在对象 A 中减去与对象 B 重合的部分。将对象 A 的修改器 Operation 设置为 Difference 模式，效果如图 4-26 所示，可以看到对象 A 将删除掉与对象 B 重合的网格，并在接口位置自动缝合接口边。

在 Blender 中一个修改器只能完成两个物体之间的 Boolean 运算，如果需要实现多个物体间的 Boolean 计算，可以为同一个物体添加多个 Boolean 修改器。将对象 B 做复制操作得到对象 C，按 G 将其移到对象 A 的左边，单击选择对象 A，在修改器中再次添加一个 Boolean 修改器，将运算

模式选择为 Union，并修改 Object 对象输入为刚才复制出来的对象 C 名称，计算效果如图 4-27 所示。这里对象 A 在与对象 B 实现 Difference 运算的基础上，又与对象 C 实现了 Union 运算。

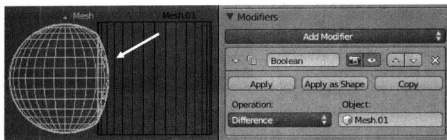


图 4-26 Difference（差集）运算

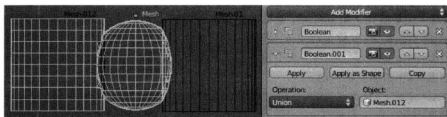


图 4-27 多物体之间的 Boolean（布尔）运算

#### 4.1.4 构造修改器

Build（构造）修改器能实现物体表面结构重建回放的特殊效果，物体的构建顺序将按照编辑模式下创建物体表面的顺序。如图 4-28 所示，为 Build（构造）修改器的参数面板。

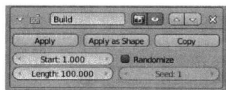


图 4-28 Build（构造）修改器的参数面板

如图 4-29 所示，从左至右是一个应用 Build（构造）修改器，完成的大圣重建动画的回放效果。

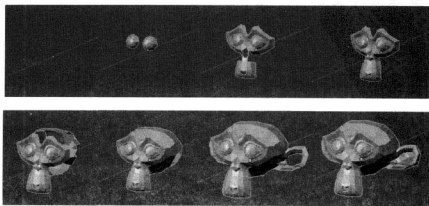


图 4-29 Build（构造）的动画效果

**Start:** 起始帧位置，用于标示构建的起始时间点。

**Length:** 变形的时间长度，单位为帧。

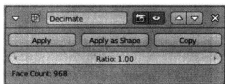
**Randomize:** 随机模式，激活后物体的表面将会采用随机的顺序回放重构造动画，而 Seed 选项则提供了随机顺序的计算方式。如图 4-30 所示，从左至右分别为 Seed 值是 1、2 和 3 的随机构造效果。



图 4-30 随机构造的动画效果

#### 4.1.5 精简修改器

使用 Decimate (精简) 修改器，可以在保持模型基本轮廓的情况下，精简掉高模上多余的点和面，快速制作出游戏等特殊应用中的低模模型。精简后的低模会使用三角面来还原模型结构，但同时 UV 会出现一部分数据的丢失。如图 4-31 所示，为 Decimate (精简) 修改器的参数面板。



**Ratio:** 比率值用于设置精简计算的缩减比，数值越大代表保留的面数越多，Face Count 显示了被精简后的模型面数量。如图 4-32 所示，从左至右分别为比率值为 1.0、0.5、0.1 的精简效果。

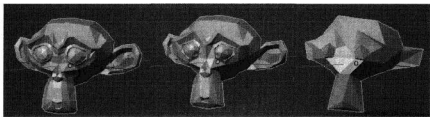
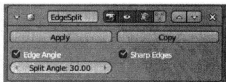


图 4-32 精简模型的效果

#### 4.1.6 硬边修改器

与平滑模型表面的效果相反，Edge Split (硬边) 修改器用于为多边形物体添加硬边效果，如图 4-33 所示，为 Edge Split (硬边) 修改器的参数面板。



**Edge Angle:** 硬边角度值，当网格上边的夹角角度小于这里设置的角度值时，修改器将不会对它作硬边处理计算。例如新建 3 个面，并分别制作出不同的边夹角度，接着使用 Shade Smooth 将它们全部做平滑处理。添加一个 Edge Split (硬边) 修改器，取消 Edge Angle 选项，效果如图 4-34 左图所示。可以看到修改器在没有角度设置时，将不会对物体做任何硬边处理。当输入硬边角度为 45 度，并开启 Edge Angle 功能时，物体上各边的效果如图 4-34 右图所示。可以看到中间和右边

的面由于边夹角度大于45度，转折的边位置被执行了硬边处理。

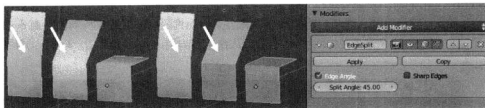


图 4-34 设置 Edge Angle 为 45 度时的硬边处理效果

**Sharp Edges:** 当激活硬边标识选项时，修改器将对所有标记为 Sharp 的边执行硬边处理，并且忽略受选项 Edge Angle 设置的角度限制。在编辑模式中，如图 4-35 左图所示，选择面正中的边，单击 Ctrl + E 后选择 Mark Sharp 将其标记为 Sharp 边，标记后的边将在编辑模式中显示为红色。接下来在修改器中打开 Sharp Edges 选项，效果如图 4-35 右图所示，可以看到标记为 Sharpe 的边全部被执行了硬边处理，并且覆盖了 Edge Angle（边角度）对硬边处理的限制。

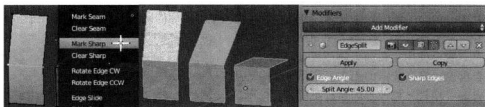


图 4-35 对标记为 Sharpe 的边执行硬边处理效果

对比这两种处理方式，可以发现，当不标记 Sharpe 边而直接对整体作硬边处理时，很难保留一些平滑面的细节，而使用硬边标识的方法，可以更灵活地对物体作特殊的硬边处理。

#### 4.1.7 遮罩修改器

Mask（遮罩）修改器能实现遮罩显示的效果，其中物体上未被遮罩覆盖的部分，在物体模式下会变得透明，而在编辑模式中则会恢复显示。如图 4-36 所示，为 Mask（遮罩）修改器的参数面板。

**Mode:** 这里提供了两种遮罩模式。

- 第一种是采用 Vertex Group（顶点组）。新建一个默认物体，单击 Tab 进入编辑模式，在点选择方式下选择一些 Loop 点，并将其定义为一个顶点组，如图 4-37 所示。

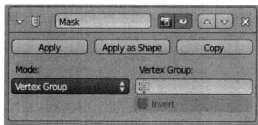


图 4-36 Mask（遮罩）修改器的参数面板

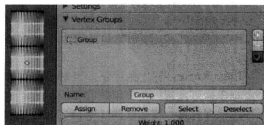


图 4-37 为物体定义顶点组

回到物体模式，为物体添加一个遮罩修改器，并选择遮罩模式为顶点组，输入顶点组的名称，效果如图 4-38 所示。这时顶点组将被计算为遮罩层，显示效果为可见，而其他部分则变为



透明。单击 Invert（反向）遮罩效果，原来的顶点组部分将被隐藏出来，而物体上其余的面将显示出来。

- 第二种模式为 Armature（骨骼）模式。首先在物体模式中单击 Shift + A 添加一个 Armature（骨骼），单击 E 将其挤压两次，如图 4-39 左图所示。选择圆柱体，接着再按住 Shift 选择骨骼，单击 Ctrl + P 建立父子关系。最后选择圆柱体，单击 Tab 进入编辑模式，分别建立三个顶点组，将骨骼对应的点分别分配给对应的顶点组，如图 4-39 所示。



图 4-38 Vertex Group（顶点组）遮罩

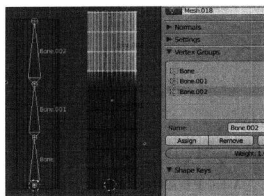


图 4-39 添加骨骼和顶点组

退出编辑模式后，选择圆柱体，更改 Mask 修改器中的 Mode 为 Armature（骨骼）模式，在旁边的输入框中填入骨骼的名称，如图 4-40 所示。



图 4-40 开启 Armature（骨骼）模式

Mask（遮罩）将仅在骨骼的 Pose（姿势）模式下生效，选择不同的骨骼时，其对应的顶点组将显示为可见，没有选择的骨骼对应的顶点组此时将显示为透明状态，效果如图 4-41 所示。

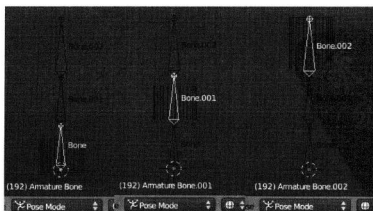


图 4-41 在 Pose 模式下选择骨骼的 Mask 效果

### 4.1.8 镜像修改器

在前面曾讲到使用关联复制来制作镜像物体，这里也可以用 Mirror（镜像）修改器来快速完成这一操作。Mirror（镜像）修改器将以物体的原点为圆心，自动对物体做本地坐标轴上的镜像操作，复制内容包括多边形网格和 UV 数据等属性。如图 4-42 所示，为 Mirror（镜像）修改器的参数面板。

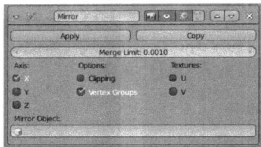


图 4-42 Mirror（镜像）修改器的参数面板

**Axis:** 复制镜像轴，分别用于设置物体的复制基准轴。如图 4-43 所示，从左至右分别是设置镜像轴为 X、Y 和 Z 轴时的复制效果，当然也可以同时选择多个坐标轴进行多重复制操作。

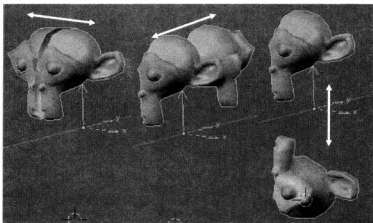


图 4-43 不同镜像轴上的复制效果

**Merge Limit:** 缝合极限，这里的数值设置了自动缝合的最小范围，当接缝处的点与镜像轴的距离小于这个数值时，原物体上的点将和镜像物体在衔接处做自动缝合。如图 4-44 左图所示，设置缝合范围为 0.01 个 Blender 单位，可以看到在点与镜像轴相距大于 0.01 时，没有缝合的效果。若把点向镜像轴稍作移动，一旦距离小于 0.01 时，接口边就被自动缝合起来，如图 4-44 右图所示。

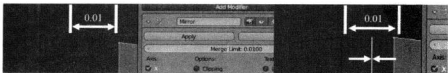


图 4-44 Merge Limit（缝合极限）

**Clipping:** 开启 Clipping（边界）功能后，修改器会限制原物体的网格不会超过镜像轴。默

认情况下,如果网格向镜像方向移动时超过了镜像轴,点的接口处会出现错位现象,如图4-45左图所示。当开启 Clipping 后,Blender 就会强约束束网格在临界处的移动,如图4-45右图所示。

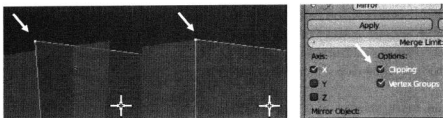


图 4-45 Clipping (边界) 效果

**Vertex Groups:** 顶点组复制,网格属性在被镜像修改的同时,将自动产生对应的镜像顶点组。这一点在人物的骨骼绑定中很实用,只需要完成一半的骨骼和顶点组设置,另一半就可以完全交给镜像修改器来完成。但是前提是需要在对顶点组命名时添加“Leg. L”或者“Arm. R”的后缀,使修改器可以识别并区分顶点组的方向。

**Textures:** 贴图镜像, U 和 V 参数分别用于设置对物体的贴图,做 UV 坐标上的反向镜像复制。如图4-46所示,左图是一个关闭贴图镜像模式的物体,可以看到右侧绘制的贴图被复制到了镜像轴的另一面。而中图是激活 U 反向的结果,右图则是激活 V 反向的结果,比较球体的本地 UV 坐标轴方向,可以看到它们分别在 U 和 V 坐标上做了反转的贴图复制计算。

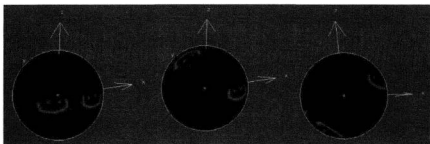


图 4-46 贴图镜像的效果

**Mirror Object:** 物体镜像,使用物体来替代镜像轴。在原物体的原心上添加一个 Empty (空) 物体,命名为 Mirror\_Control,单击大圣的镜像修改器,在 Mirror Object 中输入 Empty (空) 物体的名称。这时再使用快捷键 G 移动 Empty (空) 物体时,大圣将不再以自身原心为镜像轴,而是以 Empty (空) 物体的原心为镜像轴基准,做对称复制计算,效果如图4-47所示。

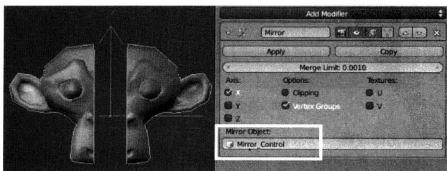


图 4-47 Mirror Object (物体) 镜像

### 4.1.9 细分修改器

Subdivision Subsurface (细分) 修改器是最常用的 Blender 修改器之一, 可用于平滑模型的表面, 将低模物体快速转换为高模物体。它的平滑计算不影响模型本身, 不会对网格做出修改, 细分效果直接显示在物体模式和渲染结果中。如图 4-48 所示, 为 Subdivision Subsurface (细分) 修改器的参数面板。

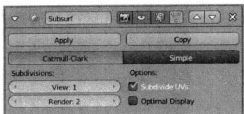


图 4-48 Subdivision Subsurface (细分) 修改器的参数面板

**Catmull-Clark/Simple:** 这是两种不同的细分计算方式, 前者会在细分的同时对物体做平滑处理, 而后者则没有平滑的效果, 只在原多边形表面上对面做细分修改。如图 4-49 所示, 左图为 Catmull-Clark 的细分效果, 而右图为 Simple 的细分效果。

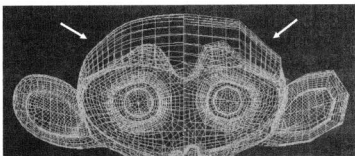


图 4-49 不同的细分计算方式

**Subdivisions:** View 为视图下可见的细分级别, Render 为渲染时物体的细分级别。在编辑状态时可以使用低等级的细分, 用于预览模型效果, 而渲染时建议调高细分等级, 以便能获得更好的平滑效果。在物体状态下, 可以使用快捷键 Ctrl + 数字键来快速地增加细分等级。

**Options:** 激活这里的 Subdivide UVs 选项能使修改器细分平滑物体上的 UV 贴图。而 Optimal Display (最优化显示) 选项, 可将使高模在透明模式中显示物体的低等级细分网格。如图 4-50 所示, 左图为 3 级细分下的线框预览, 而右图是开启最优化显示的效果。

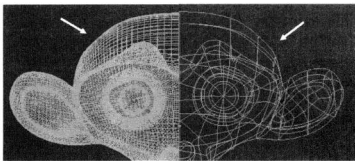


图 4-50 最优化的线框显示模式

### 4.1.10 实体化修改器

上一章曾讨论过, 如何在编辑模式下使用 Solidify 工具来实现网格实体化, 这里将介绍如何使用 Solidify 修改器来完成对物体的实体化修改。修改器对物体的实体化操作更灵活, 可以调节各项参数来控制实体化的细节, 如图 4-51 所示为 Solidify (实体化) 修改器的参数面板。

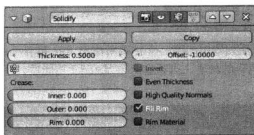


图 4-51 Solidify (实体化) 修改器的参数面板

**Thickness:** 厚度值，用于控制实体化后的网格厚度。

**Offset:** 偏移量，控制实体化后的网格与原始网格的位移偏移量。如图 4-52 所示，为厚度值为 0.5、偏移量为 0 的半球实体化效果。

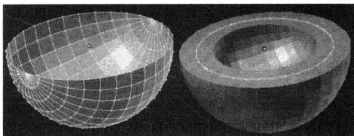


图 4-52 实体化的效果

**Crease:** 折痕选项，用于为实体化后生成的网格边缘制作尖锐的折痕效果，其中 Inner 和 Outer 分别控制内外两侧的实体边缘，而 Rim 可实现边缘的圆滑和尖锐过渡效果。如图 4-53 左图所示，为未开启 Crease 的实体物体边缘，当设置 Inner 和 Outer 分别为 1，Rim 为 0 时，实体的边缘效果如图 4-53 右图所示。

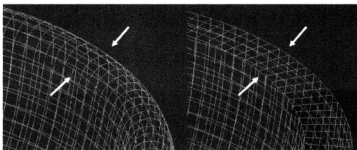


图 4-53 实体化的 Crease (折痕) 效果

**Even Thickness:** 自动调整实体化网格的厚度，当制作一些多边形的转角面时，常需要激活用于修正计算。如图 4-54 左图所示，为一个矩形转角位置的实体化效果，可以看到在转角处实体的厚度不均匀，当开启 Even Thickness 后，系统调节的效果如图 4-54 右图所示。



图 4-54 Even Thickness 调节新建实体化的厚度

**High Quality Normals:** 开启法线计算。

**Fill Rim:** 将新建的实体化网格与原物体缝合起来。

**Rim Material:** 使用材质列表中的第二个材质来渲染 Rim 接缝处的面。如图 4-55 所示, 为当前物体添加了两个不同色彩的材料, 当开启 Rim Material 功能后, 系统将自动使用第 2 个材质来渲染 Rim 面。

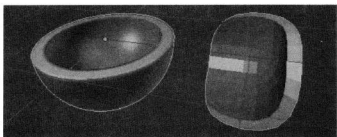


图 4-55 使用单独的材质来渲染实体化的 Rim 面

#### 4.1.11 多重解析修改器

在对物体执行雕刻操作前, 需要将模型细分至高面数的网格, 以满足雕刻对模型表面细节的需要。但是在物体的编辑模式下, 对多边形直接做物理细分会出现很多的问题, 例如当细分次数较高时, 会造成物理多边形数量超过机器承受力, 影响 Blender 的承受能力, 甚至导致死机。解决的方法如下: 为物体添加一个 Multiresolution (多重解析) 修改器, 在不改变模型物理结构的基础上, 对其做高级的多重解析细分。如图 4-56 所示, 为 Multiresolution (多重解析) 修改器的参数面板。

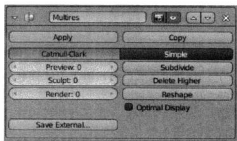


图 4-56 Multiresolution (多种解析) 修改器的参数面板

**Catmull-Clark/Simple:** 执行细分的计算模式, 和细分修改器中的计算原理类似。如图 4-57 左图所示, 为对一个 Cube 物体执行 Catmull-Clark 模式细分的效果, 如图 4-57 右图所示, 为执行 Simple 模式细分的效果。

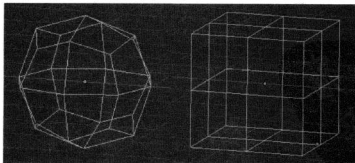


图 4-57 两种细分模式

**Subdivide:** 细分工具, 单击一次网格将被执行一次细分运算, 左边的 Preview 用于设置视图中的可视等级数, Sculpt 用于设置雕刻时的运算等级, 而 Render 则用于设置渲染时采用的运算等级。

**Delete Higher:** 删除比当前设置更高的细分等级。

**Reshape:** 拷贝其他物体的细分运算结果, 前提是两者需要有相同的点数量。

**Save External:** 将表面的置换数据保存为文件。

#### 4.1.12 UV 投影修改器

可以使用 UVProject (投影) 修改器, 将贴图图像投影仪一样, 从摄像机镜头映射至另一个物体的表面上。如图 4-58 所示, 右边为投影修改器的面板, 左边则为投影的效果。可以看到两架摄像机, 分别将同一幅贴图投影到了物体的表面上。

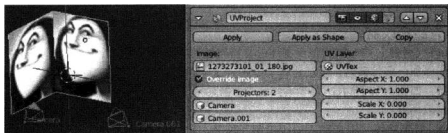


图 4-58 UV Project (投影) 修改器的参数面板

**Image:** 选择投影到物体上的贴图。

**UV Layer:** 被投影物体需要被解算出 UV, 这里选择物体的 UV 图层。

**Override Image:** 选择是否覆盖物体本身的 UV 贴图。

**Projectors:** 投影物体, 这里的数字表示允许使用几台投影物体, 并分别输入摄像机的名称。

**Aspect/Scale X/Y:** 控制水平和垂直的投影画面缩放比例。

#### 4.1.13 螺旋修改器

Screw (螺旋) 修改器可以用于制作自定义的螺旋形物体, 如图 4-59 所示, 为 Screw (螺旋) 修改器的参数面板, 和一个在 Circle 物体上应用修改器的效果。

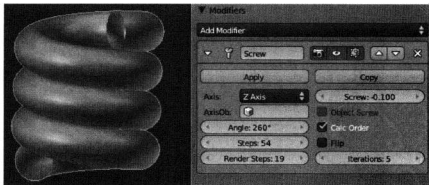


图 4-59 Screw (螺旋) 修改器的参数面板

**Axis:** 螺旋形变的基准坐标轴, 默认为 Z 轴。

**Screw:** 螺旋的深度值, 值越大螺旋体的拉伸幅度也越大。

**AxisOb:** 是否使用物体来控制螺旋形变的基准坐标轴, 通常使用 Empty (空) 物体来替代控制。

**Object Screw:** 如果使用物体来控制基准坐标轴, 是否开启物体的位移偏移量控制。

**Angle:** 螺旋形变的旋转度数设置, 值越大旋转形变的圈数也越多。

**Calc Order:** 如果是在网格物体上使用螺旋修改器, 需要激活这个选项来平滑网格表面的法线, 如果使用的是 Curve (曲线) 物体, 则没有必要激活这个选项。

**Steps:** 显示的细分步长值, 值越大新生成网格物体的表面看上去越平滑。

**Render Steps:** 渲染的细分步长值, 值越大渲染后物体的表面看上去越光滑。

**Flip:** 是否反转物体的法线。

**Iterations:** 设置螺旋形变的迭代次数, 数值越大, 螺旋体的重复次数越多。

## 4.2 形变类修改器

### 4.2.1 骨骼修改器

Armature (骨骼) 修改器类似于物体的骨骼属性模块, 用于标识骨骼的大部分基本属性。当为一个物体添加一副骨骼时, 系统将自动添加一个骨骼修改器。如图 4-60 所示, 为 Armature (骨骼) 修改器的参数面板。

**Object:** 骨骼物体, 用来输入用于控制当前物体的骨骼物体名称。

**Vertex Group:** 当该物体拥有 2 个以上的骨骼修改器时, 这里的顶点组选项可以用于区分修改器各自的影响范围。

**Bind To:** Vertex Group (顶点组) 和 Bone Envelopes (骨骼封套), 两种绑定模式均可用于控制骨骼的影响范围, 只要根据需要选择对应的模式即可。

**Quaternion:** 启用骨骼的四元数计算, 使骨骼在大角度旋转时通过四元数来计算坐标值, 以修正物体蒙皮的变形效果。Quaternion 的计算效果比常规的矩阵计算效果更好, 占用的资源空间更少, 同时还有利于坐标的插值计算。如图 4-61 左图所示为采用常规矩阵计算时的蒙皮效果, 可以看到, 如果将手臂旋转 180 度, 连接处表面会出现不规则扭曲甚至破损的效果。当开启四元数计算后, 蒙皮的矫正计算效果如图 4-61 右图所示。

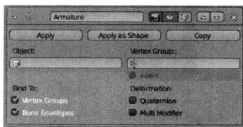


图 4-60 Armature (骨骼) 修改器的参数面板

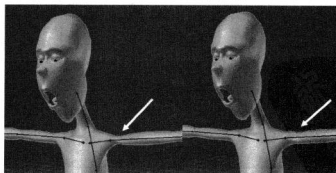


图 4-61 骨骼的 Quaternion (四元数) 计算效果

**Multi Modifier:** 在使用多骨骼时, 开启叠加过渡计算。

骨骼修改器只有针对骨架的设置, 如果需要对 Bone (骨头) 做调整, 则需要进入 Bone (骨头) 参数面板, 具体使用将在后面的章节中详细讨论。

### 4.2.2 塑性修改器

Cast (塑性) 修改器, 可将物体按照一定的形状做轮廓调整, 其效果和编辑模式中的球体化



工具操作类似,不过 Cast (塑性) 修改器提供了更多的自定义选项,可以制作出更多的特殊塑性效果。如图 4-62 所示,为 Cast (塑性) 修改器的参数面板。

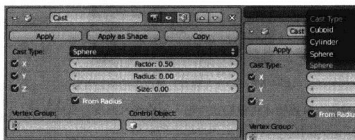


图 4-62 Cast (塑性) 修改器的参数面板

**Cast Type:** 塑性类型,单击右边的箭头可弹出类型菜单,包括 Sphere (球体化)、Cylinder (圆柱化) 和 Cuboid (方块化)。X/Y/Z 选项分别控制塑性的方向,其中针对 Cylinder 模式的变形只有 X 和 Y 两种方向选项。如图 4-63 所示,从左至右分别为 Sphere (球体化)、Cylinder (圆柱化) 和 Cuboid (方块化) 三种类型的塑性效果。

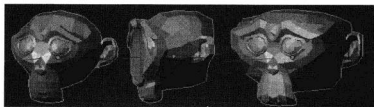


图 4-63 不同类型的塑性效果

**Factor:** 系数值,其数值大小决定了变形的幅度,设置为 0 时将保留物体的原始尺寸,修改为不同的数字有时还能得到一些很有意思的效果。如图 4-64 所示,对于同一个大圣模型,从左至右分别是 Factor 值为 -1.5、0.5 和 1.5 的变形效果。

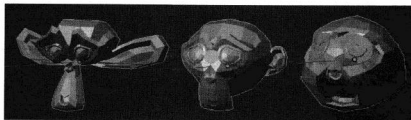


图 4-64 不同 Factor 值的塑性效果

**Radius:** 变形半径,数字代表了控制物体造型的影响范围。

**Size:** 变形的尺寸大小,设置为 0 时是自动模式。

**Vertex Group:** 仅将变形应用于指定的顶点组。

**Control Object:** 使用物体来控制变形,例如使用 Empty (空) 物体的坐标数据。

### 4.2.3 曲线修改器

Curve (曲线) 修改器能根据曲线的曲率等外形,来修改被控物体的形状造型。如图 4-65 所示,为 Curve (曲线) 修改器的参数面板。

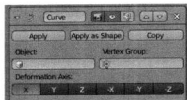


图 4-65 Curve (曲线) 修改器的参数面板

**Object:** 曲线对象, 被控物体将根据这里输入的曲线来改变自身的形状。

**Vertex Group:** 使用曲线来影响特定顶点组上的点形变。

**Deformation Axis:** 形变坐标, 用于选择曲线对物体的约束方向。

物体的细分等级越高, 形变的效果将越平滑。添加一个默认的 Plane 物体, 单击 Tab 进入编辑模式, 全选物体后, 单击 W 选择 Subdivide (细分), 重复三次细分后的效果如图 4-66 所示。

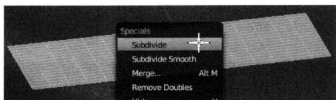


图 4-66 对物体做细分操作

接着在相同位置添加一个 Bezier 曲线, 单击 Tab 进入编辑模式, 选择节点并单击 G 修改曲线的形状, 效果如图 4-67 所示。

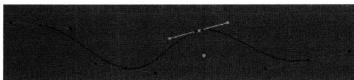


图 4-67 添加曲线物体

选择 Plane 物体, 为其添加一个 Curve (曲线) 编辑器, 在 Object 中输入曲线名称, 物体就根据曲线的弯曲程度作出了相应的形变, 效果如图 4-68 所示。

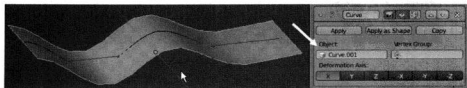


图 4-68 添加曲线修改器后的效果

#### 4.2.4 置换修改器

Displace (置换) 又称为位移映射, 它是一种使用高度贴图来制造凹凸细节的贴图技术。这种映射方式通常是根据贴图的高度定义数据, 将多边形的点在沿面法线的方向上移动一定的距离, 使得可以在较低面数的模型上表现出较丰富的深度细节。由于置换贴图可以产生高分辨率的场景模型, 而只占用较少的数据存储空间, 所以这种技术常被应用于游戏制作中。Displace

(置换) 修改器可以为物体添加实时的置换预览效果, 如图 4-69 所示, 为 Displace (置换) 修改器的参数面板。

**Texture:** 贴图文件, 选择要使用的置换贴图文件。

**Direction:** 置换的计算方向, 通常使用的方向是 Normal (法向), 也可以选择特殊的 X/Y/Z 计算方向。如果贴图是彩色的, 那么系统将会自动根据 RGB 数值来计算 XYZ 的置换深度。

**Vertex Group:** 指派修改器作用在物体表面上的顶点组。

**Texture Coordinates:** 映射坐标, 可选择将贴图映射到 UV、Object (物体) 坐标、Global (全局) 坐标和 Map (本地) 坐标系。

**Midlevel:** 中阶值, 当贴图置换值小于中阶值时, 将显示为凹效果, 大于中阶值时将显示为凸起效果。这里的置换计算公式为  $\text{Displacement} = \text{Texture Value} - \text{Midlevel}$ 。

**Strength:** 置换的强度值, 值越大置换偏移量越大, 计算公式可以表示为  $\text{Vertex Offset} = \text{Displacement} * \text{Strength}$ 。

首先添加一个默认的 Plane 物体, 单击 Tab 进入编辑模式后, 对其做多次细分处理, 并单击 Sade Smooth 执行网格平滑, 效果如图 4-70 所示。

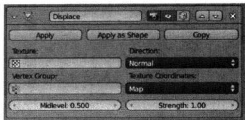


图 4-69 Displace (置换) 修改器的参数面板

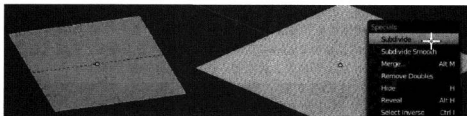


图 4-70 添加默认的 Plane 物体

转到贴图面板, 单击 Add (添加) 一个默认贴图, 选择 Type (类型) 为 Clouds (云彩), 如图 4-71 所示。

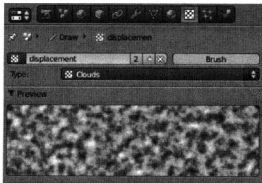


图 4-71 添加一个置换贴图

单击选择物体, 为其添加一个置换修改器, 在 Texture 中填入刚才新建的置换贴图。如图 4-72 所示, 平面根据贴图信息使用置换计算产生了凹凸的效果。

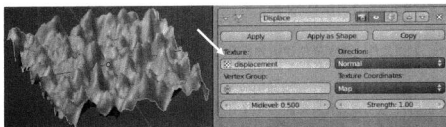


图 4-72 置换效果

### 4.2.5 晶格修改器

Lattice (晶格) 可以使物体按照晶格的外形做出对应的形变效果。如图 4-73 所示, 为 Lattice (晶格) 修改器的参数面板。

**Object:** 输入用于控制物体的晶格名称。

**Vertex Group:** 设置被晶格影响的顶点组名称。

晶格物体提供了一个十分形象的变形约束控制方式, 可以修改多种类型的物体, 包括多边形物体和曲线物体。当把它和一个物体关联起来时, 改变晶格上点的位置即可直接修改被控物体的网格形态, 但是这种修改方式并不会破坏被控物体的网格结构。晶格修改的优势在于, 无需直接对物体做结构上的修改, 就可以得到一定的形变效果, 同时工作方式也更快速而且直观。

首先在视图中的新增物体菜单下添加一个默认的晶格物体, 然后单击 Tab 进入其编辑模式, 如图 4-74 左图所示。默认的晶格由  $2 \times 2 \times 2$  个点组成, 晶格上的点越多, 控制点越多, 被控制的结构面范围越小。如图 4-74 右图所示, 为控制点的控制模式选择菜单。

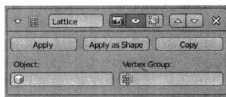


图 4-73 Lattice (晶格) 修改器的参数面板

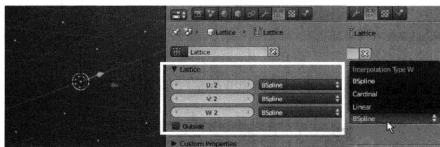


图 4-74 晶格控制参数面板

将 UVW 分别改为 4, 按 Tab 退出至物体模式, 接着在同原心的位置处添加一个球体。全选模型后使用 Shade Smooth (平滑表面), 并为其新建一个顶点组, 如图 4-75 所示。

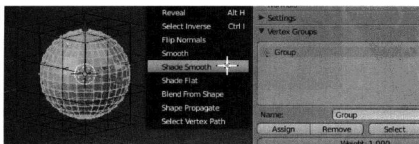


图 4-75 建立球体的顶点组

单击 Tab 回到物体模式，为球体添加一个 Lattice（晶格）修改器，在 Object 中输入晶格的名称，并在 Vertex Group 中输入建立的默认顶点组，如图 4-76 所示。

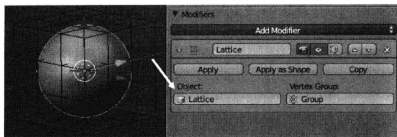


图 4-76 添加晶格修改器

接着选择晶格物体，单击 Tab 进入其编辑模式。这时会发现，当修改晶格物体的结构时，被控球体的外形将在对应晶格上的位置处产生形变效果，如图 4-77 左图所示。但是当单击 Tab 回到球体的编辑模式时就会发现，球体本身的物理结构并没有被修改，如图 4-77 右图所示。

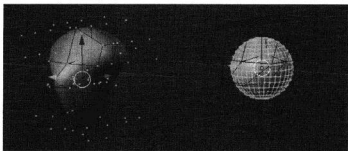


图 4-77 晶格对物体的形变操作

## 4.2.6 钩锁修改器

Hook（钩锁）修改器，用于记录物体上新添加的 Hook（钩锁）属性。晶格可以在其编辑模式下对物体执行形变操作，而钩锁则能在物体模式下辅助物体的结构形变。如图 4-78 所示，为 Hook（钩锁）修改器的参数面板。

**Object:** 钩锁物体的名称，一般来说使用 Empty（空）物体来模拟钩的功能。

**Vertex Group:** 物体中受钩锁影响的顶点组名称。

**Falloff:** 偏移量，用于控制钩对物体约束的衰减效果。

**Force:** 强度值，当物体上有多个钩作用时，可以为每个钩单独设置强度大小。

**Reset:** 重置钩对顶点组的影响。

**Recenter:** 将钩定位到光标的位置。

首先在物体模式下，按 Shift + A 键添加一个锥形物体，单击 Tab 进入其编辑模式，选择顶上的点，然后单击快捷键 Ctrl + H，弹出钩菜单，如图 4-79 所示。

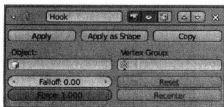


图 4-78 Hook（钩锁）修改器的参数面板



图 4-79 Hooks（钩锁）菜单

这里有两种钩选项，其中 Hook to New Object 将在此点处新建一个钩物体，而 Hook to Selected Object 则能为物体使用一个现成的钩物体。当选择 Hook to New Object 时，一个 Empty（空）物体就会出现在点的位置上，单击 Tab 退出到物体模式，可以发现系统为锥形物体自动添加了钩修饰器，如图 4-80 所示。

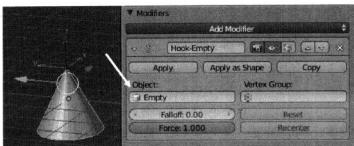


图 4-80 添加钩修饰器

这时选择 Empty 物体，按 G 做移动操作。如图 4-81 左图所示，可以看到 Empty 钩住了物体上刚才被选择的点，与 Empty 同步移动。然后改为选择圆锥物体，按 G 做移动操作。如图 4-81 右图所示，可以发现物体移动时刚才被选择的定点被固定在 Empty 位置上，无法参与物体的移动，这样也实现了类似挂钩的效果。



图 4-81 移动物体时的钩效果

#### 4.2.7 网格造型修改器

不仅可以使使用 Lattice（晶格）来改变物体的外形，也可以使用网格多边形，并配合 Mesh Deform（网格造型）修改器来修改物体的形态。如图 4-82 所示，为 Mesh Deform（网格造型）修改器的参数面板。

**Object:** 输入用于修改物体外形的网格物体名称。

**Vertex Group:** 物体中受影响的顶点组名称。

**Bind:** 绑定工具，当激活这个按钮后，网格多边形对物体的造型功能才能生效。

在物体模式中添加一个默认的大圣模型，如图 4-83 左图所示。然后单击选择大圣，按 Shift + D 复制出一个新大圣，松开鼠标后，按 S 向外稍作缩放，效果如图 4-83 右图所示。

选择新建的大圣，为其添加一个 Decimate（精简）修改器，将 Ratio 比率值降至 0.3，然后应用这个修改器的作用效果，如图 4-84 所示，这样就得到了一个低模大圣模型。



图 4-82 Mesh Deform（网格造型）修改器的参数面板

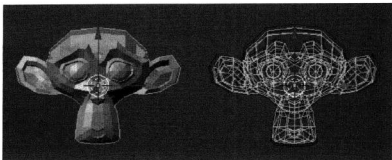


图 4-83 复制原大圣模型

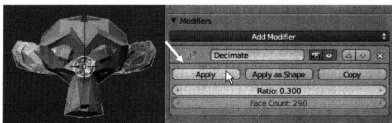


图 4-84 制作低模大圣模型

单击原高模大圣，为其添加一个 Mesh Deform 修改器，如图 4-85 所示，在 Object 中填入刚才制作的低模大圣的名称，并将修改模式设置为 Surface，单击 Bind 开始绑定控制。

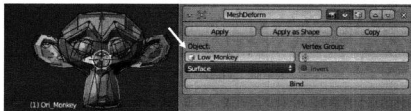


图 4-85 添加一个 Mesh Deform 修改器

选择低模的大圣，单击 Tab 进入其编辑模式，如图 4-86 左图所示。选择低模物体上的点，按 G 做移动操作，这时就可以看到高模大圣在修改器的作用下，与低模物体建立起了形变控制关系，对低模网格的修改对应地表现在被控物体的表面上，如图 4-86 中图所示。但是当退出低模物体的编辑模式，进入高模物体的编辑模式时，可以看到原物体的物理结构并没有被改变，如图 4-86 右图所示。

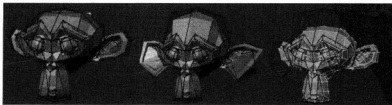


图 4-86 修改低模物体来调整原物体形态

#### 4.2.8 收缩修改器

Shrinkwrap（收缩）修改器，能将物体上的点以映射的方式收缩至目标物体的表面上。和

Deform 类的其他修改器一样,它也只能被应用显示在物体模式下,并且不会改变原物体的自身结构。如图 4-87 所示,为 Shrinkwrap (收缩) 修改器的参数面板。

**Target:** 选择收缩的目标物体。

**Offset:** 与目标物体之间的距离偏移量,当值为 0 时物体将紧贴目标物体。

**Subsurf Levels:** 细分等级。

**Mode:** 模式,这里有两种收缩模式,一种是 Nearest Vertex/Surface Point,用于将物体收缩至最近的点或表面点的位置,另一种是 Project 完全映射模式。

**Keep Above Surface:** 将物体始终控制在收缩目标物体的上方,使其不会出现相互穿越的现象。

在物体模式中添加一个默认的 Plane (平面),单击 Tab 进入其编辑模式,将其细分 5 次。然后打开衰减编辑模式,为平面塑造一些起伏效果,并使用 Smooth (平滑) 物体的表面,效果如图 4-88 所示。

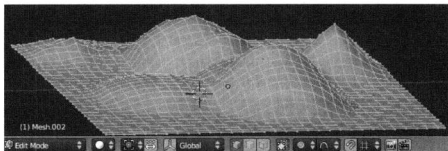


图 4-88 制作起伏的平面效果

接着在物体的正上方添加一个贝济埃曲线,单击 Tab 进入编辑模式,对其形态做一些调整,如图 4-89 左图所示。然后在曲线的修改器面板中为其添加一个 Shrinkwrap 修改器,在 Target 中输入平面的名称,设置 Offset (偏移量) 为 0.01,选择模式为 Project (映射),将映射轴设定为 Z 方向,方向为 Negative (负) 方向,如图 4-89 右图所示。

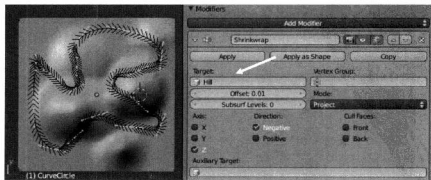


图 4-89 为曲线添加 Shrinkwrap (收缩) 修改器

单击 Tab 回到物体模式,可以看到曲线被完全收缩映射至目标物体的表面上,如图 4-90 左图所示。当把模式修改为 Nearest Vertex 时,打开平面的网格显示,可以看到曲线的吸附方式发

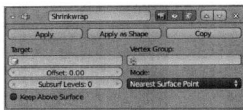


图 4-87 Shrinkwrap (收缩) 修改器的参数面板



生了改变，它不再是根据形态做映射收缩，而是将曲线上的每个节点都吸附在物体的 Vertex（点）上，如图 4-90 右图所示。

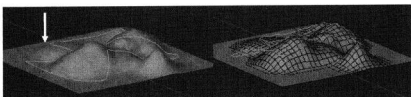


图 4-90 收缩到目标物体上的效果

#### 4.2.9 快速形变修改器

SimpleDeform（快速形变）修改器，提供了一些精确的形变修改功能。类似地，它也只能被应用显示在物体模式下，并且不会改变原物体自身的物理结构。如图 4-91 所示，为 SimpleDeform（快速形变）修改器的参数面板。

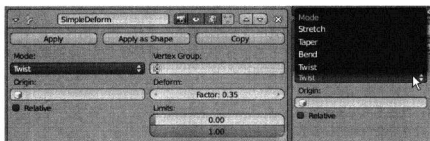


图 4-91 SimpleDeform（快速形变）修改器的参数面板

**Mode:** 模式，这里的模式菜单列举了所有的形变方式。

**Origin:** 使用自定义的物体来控制形变的原点位置。

**Relative:** 选择使用控制物体的相对坐标。

**Deform:** 形变量设置参数。

**Limits:** 形变范围设置参数。

首先在物体模式中添加一个 Cube 物体，然后单击 S + Shift + Z 对物体做压缩操作，并细分 4 次。接着在修改器面板中为其添加一个 SimpleDeform（快速形变）修改器，设置其 Deform（形变量）为 1，具体参数如图 4-92 所示。

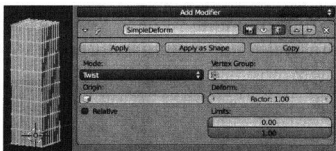


图 4-92 为物体添加 SimpleDeform 修改器

接下来退出编辑模式，开始修改形变的模式。如图 4-93 所示，从左至右分别为 Twist（扭

转)、Bend (弯曲)、Taper (锥化) 和 Stretch (拉伸) 的变形效果。如果单击 Tab 进入物体的编辑模式, 可以发现原物体的物理结构依然没有被修改, 如图 4-93 最右图所示。

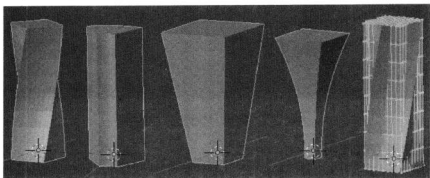


图 4-93 物体的形变效果

#### 4.2.10 平滑修改器

Smooth (平滑) 修改器和编辑模式中的平滑功能效果类似, 唯一的区别是编辑模式中的平滑操作是一次性的, 而 Smooth (平滑) 修改器可以对平滑结果做多次实时的微调。如图 4-94 所示, 为 Smooth (平滑) 修改器的参数面板。

**Factor:** 设置平滑力度参数。

**Repeat:** 设置平滑次数参数。

**Axis:** 设置平滑约束坐标参数。

在物体模式中添加一个大圣物体, 为了让平滑效果更明显, 单击 Tab 进入编辑模式, 对其分别执行 Subdivide Smooth (平滑细分) 和 Shade Smooth (表面平滑) 操作, 效果如图 4-95 所示。

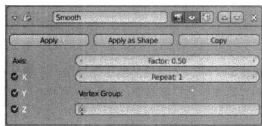


图 4-94 Smooth (平滑) 修改器的参数面板

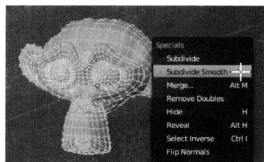


图 4-95 对大圣做平滑细分

将 Repeat 设置为 4, 如图 4-96 所示, 从左至右分别为设置 Factor 数值为 -0.5、0.5 和 1.5 下的平滑效果。



图 4-96 不同 Factor 数值下的平滑效果

### 4.2.11 波浪修改器

Wave (波浪) 修改器能通过置换的计算方式, 在物体表面上制作出类似海洋波浪的效果。如图 4-97 所示, 为 Wave (波浪) 修改器的参数面板。

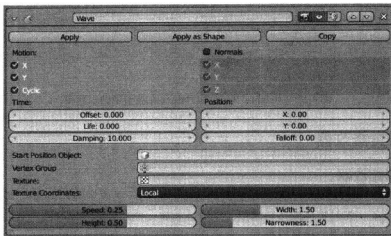


图 4-97 Wave (波浪) 修改器的参数面板

**Motion:** 设置波浪的方向, 其中 X 和 Y 分别用于约束波浪的走向。

**Normals:** 开启网格物体在法线方向上的计算效果。

**Time:** 用于设置波浪的生命周期。

**Position:** 波浪的原心位置, 如图 4-98 所示, 为在一个普通平面上制作的一个最简单的默认波浪效果。

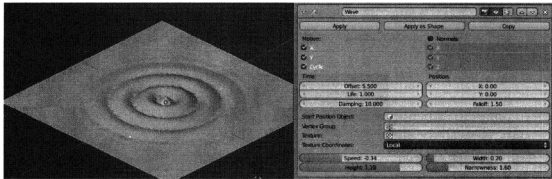


图 4-98 一个默认的标准波浪

除了静态的默认设置之外, Wave (波浪) 修改器还提供了用于制作波浪动画的设置参数。

**Start Position Object:** 选择自定义物体对象来控制波浪的变化。

**Texture/Texture Coordinate:** 使用材质来控制波浪的置换细节, 其中 Coordinate (坐标系) 可用于选择贴图的投影方式。

**Speed:** 波浪的速度, 单位为每帧上前进的 Blender 单位距离。

**Height:** 波浪的高度, 同样也是以 Blender 单位为基础。

**Width:** 波浪的宽度, 为两个浪尖之间的距离。

**Narrowness:** 脉冲基准单位设置, 每 4 个单位为一次完整的波浪。

如图4-99所示,为Height、Width和Narrow三个参数在波浪中的示意图,理解这些参数的关系即可调节出理想的波浪造型了。

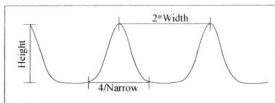


图4-99 Height、Width和Narrow在波浪中的示意图

68

## 4.3 模拟类修改器

### 4.3.1 布料修改器

Cloth (布料) 修改器为布料模拟的辅助修改器,它用于更改当前修改器与其他修改器之间的运算顺序。例如,先对物体做布料模拟,然后再添加一个平滑细分修改器,这样不仅能获得一个高效而快速的运算过程,还可以得到一个满意的模拟结果。如图4-100所示,为Cloth (布料) 修改器的参数面板。



图4-100 Cloth (布料) 修改器的参数面板

### 4.3.2 碰撞修改器

同Cloth (布料) 修改器一样,Collision (碰撞) 修改器也是物体模拟修改器中一个辅助修改器。如图4-101所示,为Collision (碰撞) 修改器的参数面板,其具体使用将在后面详细讨论。



图4-101 Collision (碰撞) 修改器的参数面板

### 4.3.3 爆破修改器

和Build (构造) 修改器的效果相反,Explode (爆破) 修改器用于制作一种类似物体爆炸而产生的支离破碎的效果。如图4-102所示,为Explode (爆破) 修改器的参数面板。

**Vertex group:** 参与爆破的顶点组名称。

**Split Edges:** 激活后,每个爆破碎片将被执行二次细分计算,用于产生更多更小的碎片。

**Unborn:** 设置未爆破物体是否可见。

**Alive:** 设置爆破碎片是否可见。

**Dead:** 设置超过生命周期的碎片是否还可见。

**Refresh:** 重新计算爆破过程。



图4-102 Explode (爆破) 修改器的参数面板

使用Explode (爆破) 修改器之前必须先为物体添加一个粒子系统,首先添加一个默认球

体, 并对其做平滑处理。然后在修改器面板中添加一个粒子属性, 保持默认的设置, 接着为其添加一个 Explode (爆破) 修改器, 并激活 Split Edges 选项, 如图 4-103 所示。

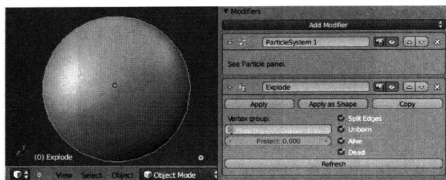


图 4-103 为球体添加 Explode (爆破) 修改器

单击 Alt + A 开始计算爆破, 如图 4-104 所示, 从左至右分别是第 0 帧, 第 15 帧, 第 30 帧, 第 45 帧, 第 60 帧, 第 75 帧, 第 90 帧和第 100 帧的爆破效果。

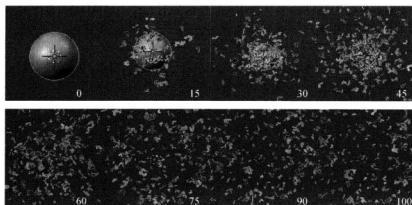


图 4-104 爆破效果

### 4.3.4 流体修改器

Fluidsimulation (流体) 修改器和 Cloth (布料) 修改器一样, 是一个辅助的状态修改器, 用于控制流体模拟与其他修改器之间的叠加关系。如图 4-105 所示, 为 Fluidsimulation (流体) 修改器的参数面板。



图 4-105 Fluidsimulation (流体) 修改器的参数面板

### 4.3.5 粒子实体修改器

ParticleInstance (粒子实体) 修改器可将目标物体的发射粒子替换为当前的实体物体。如图 4-106 所示, 为 ParticleInstance (粒子实体) 修改器的参数面板。

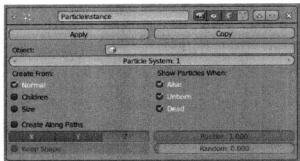


图 4-106 ParticleInstance (粒子实体) 修改器的参数面板

**Object:** 输入目标物体的名称, 这个物体一定要开启粒子发射系统属性。

**Particle System:** 当对象物体拥有多个粒子系统时, 这里可输入选择使用的粒子系统。

**Create From:** 选择粒子实体的替换方式, 其中 Normal 将选择法向的粒子, 也就是替换非子粒子的粒子, 而 Children 选项将替换发射出来的子粒子。当 Size (尺寸) 属性被激活后, 系统将使用粒子本身的尺寸来控制替换后的新实体粒子大小。

**Show Particles When:** 这里列举了替换后粒子的显示方式, 其中 Alive 用于显示发射中的粒子, Unborn 将显示未发射的粒子, 而 Dead 则显示超过了生命周期的粒子。

**Create Along Paths:** 将粒子的实体形态约束到粒子的发射方向上。

在物体模式下单击 Shift + A 添加一个 Plane 物体, 然后进入其编辑模式, 对其做 3 次细分修改, 并在修改器面板中为其添加一个粒子系统修改器, 如图 4-107 所示。



图 4-107 为 Plane 添加默认粒子发射系统

接着退回到物体模式, 再次单击 Shift + A 添加一个大圣, 将它缩小一点, 再使用 Shade Smooth 做平滑处理。单击修改器面板为其添加一个 ParticleInstance (粒子实体) 修改器, 如图 4-108 所示。在 Object 中输入刚才新建的 Plane 物体名称。

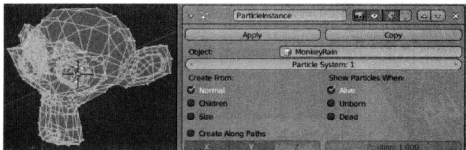


图 4-108 添加大圣物体和 ParticleInstance (粒子实体) 修改器

回到物体模式下, 单击 Alt + A 就可以预览动画了, 这时可以看到一群大圣在平面上跳跃了起来, 如图 4-109 左图所示。如果屏蔽掉修改器的作用, 就可以看到没有经过物体实例替换的原

粒子效果，如图 4-109 右图所示。

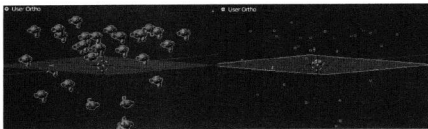


图 4-109 使用物体实例化粒子的效果

### 4.3.6 粒子系统修改器

ParticleSystem（粒子系统）修改器，是粒子系统的辅助状态修改器。如图 4-110 所示，为 ParticleSystem（粒子系统）修改器的参数面板。当为物体添加一个粒子系统修改器时，系统将默认为物体添加一个新的粒子系统。（粒子系统）修改器只提供一个状态显示，而没有任何详细的粒子参数设置。



图 4-110 ParticleSystem（粒子系统）修改器的参数面板

### 4.3.7 烟雾/软体模拟修改器

Smoke（烟雾）模拟修改器和 Softbody（软体模拟）修改器，分别为烟雾模拟和软体模拟的状态修改器。如图 4-111 所示，为 Smoke（烟雾）模拟修改器和 Softbody（软体模拟）修改器的参数面板。同粒子系统修改器一样，这两种修改器只提供了一个属性的状态显示，具体的参数设置需转自物体的物理设置面板。

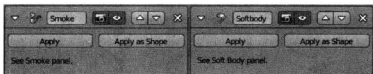


图 4-111 Smoke（烟雾模拟）修改器和 Softbody（软体模拟）修改器的参数面板

## 第 5 章

# 灯 光

无论是在电影还是摄影的过程中，灯光都是决定画面品质的关键因素之一，合理的用光能得到极富震撼力的视觉效果。CG 的灯光和传统的摄影布光方法不一样，它使用了一些特定的手法来模拟真实光照，营造虚拟的现实环境。Blender 可以模拟真实世界中的大部分光线效果，包括自然光和人工光。在本章中，将简单介绍与灯光物体相关的基础知识。

### 5.1 光线的属性

优秀的灯光设计能激活一个即使很简单的模型，而一个呆板的灯光照明则可能会毁掉一个精心搭建的场景。如何布灯是一个难以量化的问题，因为这里没有可量化的参数来快速调节，往往需要靠经验和大量的实验来逐渐修改和优化。

灯光需要根据不同的场景来做灵活搭配，为了获得更好的效果，艺术家们通常会利用照明，在不同的方向上打上多盏灯。如图 5-1 左图所示，为电影《Sintel》中的一个场景布光设计，可以看到灯光师在各个方向分别布置了点光、聚光灯和面光源，它们将分别实现主光、补光和间接光照的功能。如图 5-1 右图所示，为渲染后的效果。



图 5-1 电影《Sintel》中的灯光设计与渲染效果

在为 CG 场景调节灯光效果时，光线的色彩和强度效果可能会受到其他因素的影响，导致测试和实际渲染的效果出现偏差。例如显示器类型，LCD 的色彩相比于 CRT 会偏灰，色彩鲜艳度和饱和度上也不如 CRT。但是即使是 CRT 显示器，如果使用了非线性标准的色彩显示配置，也会导致显示和输出之间出现色彩误差。如果选择了不同的文件格式来输出测试图，也可能会导致颜色的丢失，例如 JPEG 格式会对图片做有损压缩处理。



## 5.1.1 光源属性

影响照明效果的 Source（光源）属性包括光源的强度、光线的颜色和光照下的阴影。

### 5.1.1.1 Energy（光源强度）

Energy（光源强度）又称为光强，是指光源的发光强度，同时也可用于表示物体被照明的程度。当光照强度不够时，场景就会因为照明不足而显得黑暗，如图 5-2 左图所示。但是如果光强值太高，又会导致场景曝光过度，使物体表面因为较高的亮度而丢失颜色等细节，如图 5-2 右图所示。所以恰当的灯光强度是保证画面清晰的基础，合理的效果应该如图 5-2 中图所示。



图 5-2 不同灯光强度的效果

如图 5-3 所示为灯光的 Energy 光照强度参数菜单，通过修改控制栏中的数字来调节光强大小。



图 5-3 灯光的 Energy（光照强度）参数菜单

### 5.1.1.2 Color（光线颜色）

Color（光线颜色）就是使用不同的颜色来改变照明光线，使每个光源都拥有一个彩色的色彩，从而为场景添加更丰富的色彩细节。但是光照颜色只有在合适的光照强度下才能区分，例如在使用高强度和高饱和度设置的灯光照明时，就会很难区分黄色和白色。

如图 5-4 所示，为灯光的色彩面板，单击色板后会弹出拾色器，使用 LMB 即可选择颜色和其明亮度，灯光的默认颜色为白色。

光线的颜色常被用来表现不同的环境效果，烘托出特殊的气氛。例如制作一个简单的大圣机器人，使用蓝色和白色为主要光源，这样的场景看上去显得十分平静，如图 5-5 左图所示。当改变场景中光源的色彩，设置主光源为红色、辅助光源为绿色时，场景一下就增加了不少紧张的气氛，画面也兴奋起来，如图 5-5 右图所示。

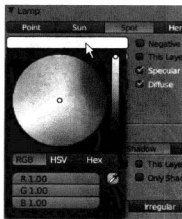


图 5-4 灯光色彩面板

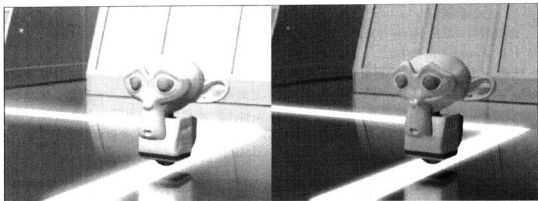


图 5-5 光源色彩对环境气氛的影响

### 5.1.1.3 Shadow (光照阴影)

物体在不同光线下的阴影效果是不同的，例如强光下的物体阴影边界应该是清晰而尖锐的，而散射光下的物体阴影边界则是柔和的。阴影的颜色也是可以改变的，有时特殊的颜色还能营造出一些有趣的氛围。Blender 中可以手动调节物体的光照阴影效果，如图 5-6 所示，为灯光的阴影颜色参数面板。在灯光的 Shadow (阴影) 子菜单中，找到颜色面板，单击色板后即可弹出拾色器，使用 LMB 选择颜色，并调节其明亮度，阴影的默认颜色为黑色。

如图 5-7 所示，左图是在南瓜内装上一个黄色光线黑色阴影的灯光效果，这样的画面看上去略显呆板。那么将阴影修改成红色时，效果如图 5-7 右图所示，南瓜下面的阴影变成了鲜红色，画面就在色彩的带动下活跃了起来。



图 5-6 阴影颜色参数面板

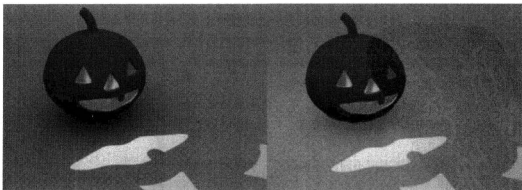


图 5-7 使用不同颜色阴影的效果

## 5.1.2 光线衰减

在大气中，由于能量转换和空间散射等原因，当光线传播至一定距离时，会出现光能衰减的

现象。所以在摄影等实际应用中,就使用光线衰减来定义光线随光源的远而产生的照明能力衰减效果。光线衰减常用于减弱或柔滑阴影的边缘效果,以及改变主体与背景之间的层次关系。现实世界中的光强是随着距离的增加按照反向平方衰减的,但是在3D环境中需要手动调节光线的衰减算法,以得到令人满意的效果。

以 Spot (聚光灯) 为例,在灯光的控制面板中找到 Falloff (衰减) 选项,如图 5-8 所示。单击下边的箭头可以弹出衰减类型菜单,这里包括 Lin/Quad Weighted (线性/二次曲线模式)、Custom Curve (自定义曲线模式)、Inverse Square (反向二次方模式)、Inverse Linear (反向线性模式) 和 Constant (常值模式)。

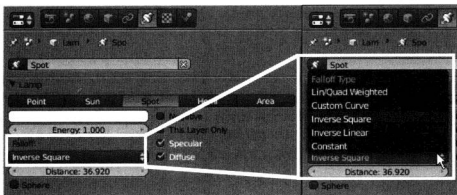


图 5-8 衰减控制模式的参数面板

#### 5.1.2.1 Linear/Quadratic (线性/二次曲线模式)

Linear (线性) 和 Quadratic (二次曲线) 属性的取值范围为 0~1, 当 Linear 为 0 而 Quadratic 为 1 时,说明灯光将完全使用二次曲线来控制衰减。如图 5-9 所示,为线性/二次曲线模式的参数面板。

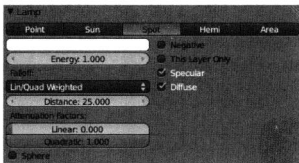


图 5-9 线性/二次曲线模式的参数面板

这里要注意,光线的衰减速度还和 Distance (光程距离) 密切相关,Distance 选项定义了光线的有效距离。如图 5-10 所示,Spot 光源的光程为从锥形顶到底部的 20 个 Blender 单位。

如果选择衰减模式为纯 Linear (线性),那么当光线从光源前进 20 个单位时,按照 Linear (线性) 的衰减比例,Energy (光照强度) 就会变得只有原来的 1/2。可以使用如下公式来计算在 Linear (线性) 衰减模式中,光照强度与光程距离的关系。其中  $I$  为最后得到的光照强度,  $E$  为强度设置值,  $D$  为距离设置值,  $L$  为线性衰减值,  $R$  为当前位置离光源之间的距离。

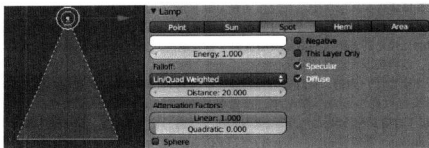


图 5-10 Distance (有效光程)

$$I = E \times \frac{D}{D + L \times R} \quad (5-1)$$

对于 Quadratic (二次曲线) 模式, 它的衰减方式更接近于真实世界。与 Linear (线性衰减) 模式相比, 光照强度的衰减会根据距离按照二次曲率来减弱, 离光源越远, 衰减得也越快, 但是在有效光程距离内, 它的衰减会比线性衰减更缓慢。使用如下公式来计算在 Quad (二次曲线) 衰减模式下光照强度与距离的关系。其中  $I$  为最后得到的光照强度,  $E$  为强度设置值,  $D$  为距离设置值,  $Q$  为二次曲线衰减值,  $R$  为当前位置离光源之间的距离。

$$I = E \times \frac{D^2}{D^2 + Q \times R^2} \quad (5-2)$$

当两种模式的值都不为 0 时, Blender 将会重叠两种衰减效果, 根据两种模式的预设值和光程距离来决定此刻光线强度会倾向哪一种衰减特性。如果两个值都为 0, 那么光照强度将不会根据距离而衰减。但这并不意味着光线强度不会降低, 由于散射的角度原因, 光线还是会因为向外无限地扩散而变暗。使用如下公式来计算两种模式叠加的效果。

$$I = E \times \frac{D}{D + L \times R} \times \frac{D^2}{D^2 + Q \times R^2} \quad (5-3)$$

以有效光程为 20 的灯光为例, 如图 5-11 所示, 为其光线强度衰减和光程距离的关系图。其中, a 线条表示纯线性的衰减 (Linear = 1, Quad = 0), b 线条表示纯二次曲线的衰减 (Linear = 0, Quad = 1), c 为线性和二次曲线的叠加效果 (Linear = Quad = 0.5), d 为无衰减的效果 (Linear = Quad = 0)。图中的横坐标为光线的传播距离值, 纵坐标则为光线的强度值。如果开启了 Sphere (范围) 选项, 无论设置为何种类型的衰减算法, 光强值都会在光程为 20 的时候被归零, 因为 Sphere 可以将光照强度在光程范围的尽头处置为 0。

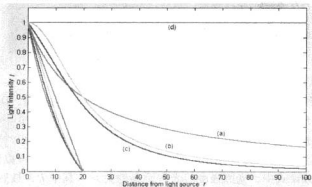


图 5-11 不同模式下的光照强度衰减图

### 5.1.2.2 Custom Curve (自定义衰减曲线)

Custom Curve (自定义衰减曲线) 可以更灵活地控制衰减方式, 当选择这种衰减模式后, 灯光面板中会出现一个新的曲线控制面板。一般情况下, 灯光的衰减方向都是由光源指向远方, 光强值从最大值在向外发散时将逐渐衰减降低, 而自定义曲线则可以调节出特殊的光源衰减起点和反向的衰减方向。如图 5-12 所示, 为自定义衰减曲线的参数面板。

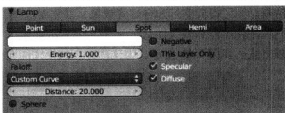


图 5-12 自定义衰减曲线的参数面板

衰减曲线图拥有两条坐标轴, 横坐标为光线的前进距离, 纵坐标为光源的强度值, 使用 LMB 就可以在曲线上添加新的节点。横坐标上, 最左边的原点为初始光源位置, 越往右越接近光线有效距离的末端。纵坐标上, 最下面的原点处光源强度值为 0, 坐标轴的顶部表示强度的最大值。如图 5-13 左图所示, 为默认自定义曲线, 它的衰减方式也近似于 Linear (线性) 衰减的效果。如图 5-13 右图所示, 为衰减的光影范例。

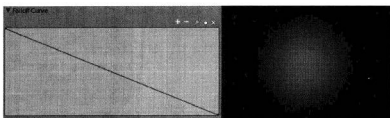


图 5-13 默认的衰减曲线与渲染效果

改变曲线形状, 添加新的节点来重定位光源的起始位置和光源初始值, 如图 5-14 左图所示。根据曲线示意, 光照将从最远向光源位置衰减, 效果如图 5-14 右图所示。

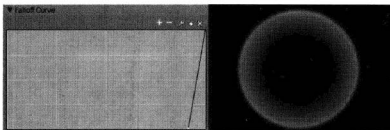


图 5-14 光照从远至近衰减的渲染效果

当然也可以添加更多的曲线形态, 用来模拟出一些特殊而有趣的效果, 如图 5-15 所示。

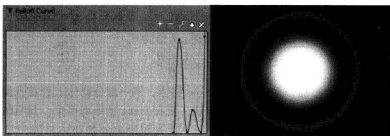


图 5-15 自定义的衰减渲染效果

### 5.1.2.3 Inverse Square/Linear (反向二次曲线/反向线性化衰减)

反向二次型曲线相比于二次型曲线模式，反转了衰减曲线，从而使光照在边缘位置显得更清晰。为了得到路灯和台灯等效果，常需要选择使用反向二次型作为灯光光线的衰减模式。而反向线性化则能和纯线性化衰减产生同样的效果，唯一的区别是它在面板中取消了二次曲线的叠加方式。如图 5-16 所示，为这两种衰减模式的面板。

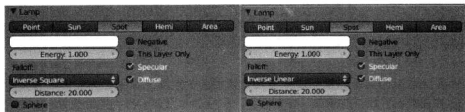


图 5-16 反向二次曲线/反向线性化衰减的参数面板

### 5.1.2.4 Constant (常值) 模式

当选择衰减模式为常值时，其效果相当于  $\text{Linear} = \text{Quad} = 0$ 。当需要制作太阳或天空效果时，可以选择这种无衰减模式，而 Blender 中内置的 Sun 和 Hemi 灯光，也都默认使用了无衰减的照明计算。如图 5-17 所示，为常值模式的灯光参数面板。

### 5.1.2.5 Sphere (范围)

前文曾提到当开启 Sphere (范围) 功能时，光照强度会在光程有效距离的末端归零。这个 Sphere (范围) 选项只针对于 Point 和 Spot 灯光，用于将光照强度控制在有效光程范围内。当开启 Sphere 功能时，光源上会出现一个半径为光程设置值大小的虚线球体。如图 5-18 所示，光照将被约束在这个球体内，超过这个范围的物体将不受这盏灯光的照明。



图 5-17 常值模式的灯光参数面板

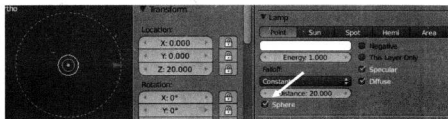


图 5-18 Sphere 开启后的光源效果

激活了 Sphere 的灯光，会在原衰减曲线值上叠加一个斜率系数，使光照强度值能快速收敛至 0。计算公式如下，其中  $I'$  为叠加后的光照强度值， $I$  为未叠加的光照强度值， $D$  为距离设置值， $R$  为当前位置离光源之间的距离。当  $R > D$  时， $I'$  值为 0。

$$I' = I \times \frac{D - R}{D} \quad (5-4)$$

如图 5-19 左图所示，添加一系列的默认 Plane 物体，将一部分物体放置在光照范围外。当未启用 Sphere 功能时，渲染结果如图 5-19 中图所示，所有 Plane 物体都可见，其明暗度受光线的衰减而有规律的降低。当为灯光开启 Sphere 功能后，在光照距离外的物体将因为没有光照，而完全变暗至不可见，如图 5-19 右图所示。

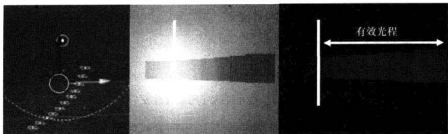


图 5-19 开启 Sphere 选项的效果

### 5.1.3 贴图灯光

当为一个灯光添加材质贴图后,灯光就会根据贴图中的选项,在照明区域产生贴图的花纹或镂空的效果。如图 5-20 左图所示,首先选择灯光物体,然后转入贴图面板。单击 Add 按钮添加一个贴图,将 Type (类型) 选择为 Voronoi 内置花纹。在 Influence (影响) 设置中修改颜色为蓝色,如图 5-20 右图所示。

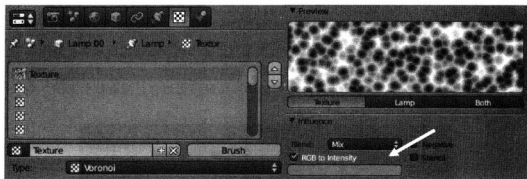


图 5-20 为灯光添加贴图材质

设置一个简单的场景,单击渲染快捷键 F12 后,就可以看到在场景中,所有被灯光覆盖的物体都印上了花纹,效果如图 5-21 左图所示。而如图 5-21 右图所示,则为取消灯光贴图的原始场景效果。



图 5-21 使用带贴图的灯光渲染效果

### 5.1.4 体积光

Volume Light (体积光) 是灯光与大气相互作用的一种有形光束效果,例如海上的灯塔和电筒发射的光线。Blender 中仅有 Spot 可开启体积光效果,如图 5-22 所示,为 Spot (聚光灯) 的灯光参数面板。

**Size:** 光照范围,值越大灯光的照射区域越广。如图 5-23 所示,左图为 Size 为 45 度时的灯光形状,右图为 Size 为 20 度时的灯光形状,夹角和对应的底部面积形象地说明了光照的有效范围。

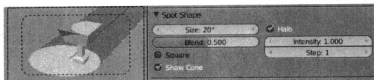


图 5-22 聚光灯的灯光参数面板

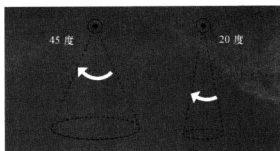


图 5-23 Size 光照范围设置

**Blend:** 边缘融合值, 值越大内光照边缘与非光照边缘之间的过渡范围越大。如图 5-24 所示, 左图为 Blend 值为 0.5 时的灯光形状, 可以看到光圈的位置多了一个内虚线环。当 Blend 值越大时, 这个环越小, 对应在内环与外环之间的阴影将显得更虚化, 渲染效果如图 5-24 右图所示。

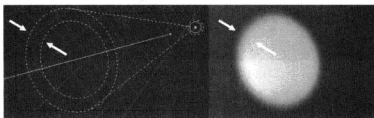


图 5-24 Blend (边缘融合值)

当物体位于虚线环区域内时, 表面将产生渐进的光影效果, 而外虚线环上的阴影边缘则较为清晰硬朗。如图 5-25 左图所示, 为 Blend 为 0 的灯光效果, 右图则为 Blend 0.5 时的物体光影效果。

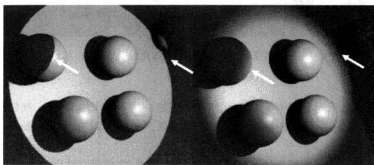


图 5-25 开启 Blend 后的渐进阴影效果



**Square:** 矩形光照，默认光照区域为圆形，单击 Square 后可将光照范围修改为矩形。

**Show Cone:** 在 3D 视图场景中显示体积光的照明范围。

**Halo:** 光晕，开启后可激活体积光效果。其中 Intensity 为体积光部分的强度，值越大光束越明亮。Step 定义了体积光中的体积阴影等级，等级越高阴影的质量反而越低，渲染时间也会变短。如图 5-26 左图所示，为 Intensity 为 1，Step 为 1 的渲染效果，当增大 Intensity 值后，光束亮点将变强甚至曝光过度，如图 5-26 右图所示。

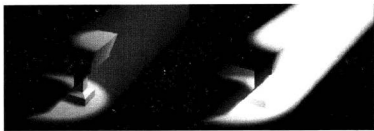


图 5-26 Halo（光晕）下的体积光效果

### 5.1.5 灯光的常规控制面板

如图 5-27 所示，为灯光的常规控制参数面板。

**Negative:** 反向照明效果，激活后灯光不会照亮这个范围内的物体，而是根据灯光强度和衰减等属性反向“照暗”这个区域内的物体。

**This Layer Only:** 仅照亮和灯光在同一个图层的物体。

**Diffuse:** 开启漫反射照明效果，否则物体将失去漫反射的光照效果。

**Specular:** 开启高光效果，关闭后物体将无高光照明效果。

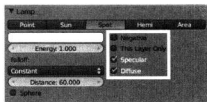


图 5-27 光线的常规控制参数面板

## 5.2 阴影类型

当物体挡住了光线的传播方向，光线就会在物体无法穿过的背面区域形成影子。影子能为场景增加更多的层次和立体感，没有影子的物体看上去往往缺乏重量感和真实感，因此在 CG 中合理地调整阴影是提高渲染质量的最重要方法之一。

由于阴影在 3D 环境中不是自动产生的，需要根据灯光和环境的属性，使用 Ray-Tracing（光线追踪）和 Buffer Shadow（缓冲阴影）等方式来创建物体的阴影。除了配置灯光，还需要在显示阴影的物体材质选项中，激活其接受阴影的显示选项，并在渲染器面板中开启对阴影的计算选项，如图 5-28 所示。

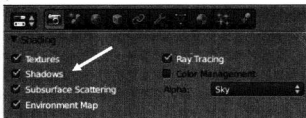


图 5-28 开启阴影的计算选项

对于上述两种类型的阴影，除 Hemi 外所有灯光都支持 Ray-Tracing Shadow（光线追踪阴影），而只有 Spot 灯光支持 Buffer Shadow（缓冲阴影）。

## 5.2.1 光线追踪阴影

### 5.2.1.1 Ray-Tracing（光线追踪）的原理

Ray-Tracing（光线追踪）是一种特殊的图形算法，系统通过跟踪与表面发生交互作用的光线，从而得到光线经过路径的计算模型。物理世界中的光都是从光源发出向前传播，投射到物体上后会被吸收或者反射，最后进入人眼，让人看到“渲染”后的真实世界。为了在三维环境中精确地模拟光线路径，并且减少计算量，计算机采用反向递归的数学算法，让光线的追踪始于眼睛而不是光源。如图 5-29 所示，假设渲染的画面是一个透明的窗口，那么光线追踪就是从镜头和画面出发，使用反向的追踪方式，来计算场景内所有用于照明虚拟世界内物体的光线。

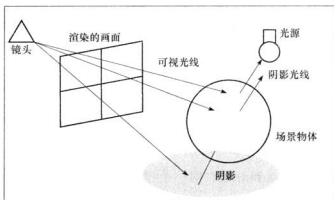


图 5-29 Ray-Tracing（光线追踪）的原理示意图

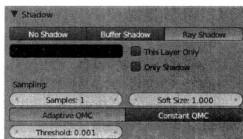
### 5.2.1.2 Ray-Tracing（光线追踪）的参数面板

当添加一个 Spot 灯光后，单击下面的 Shadow 菜单即可找到 Ray Shadow（阴影）菜单，如图 5-30 所示。可以指定阴影的颜色，默认为黑色，单击色板可在弹出的拾色器内选择其他颜色。

**This Layer Only:** 只计算和灯光在同一个层中的物体阴影。

**Only Shadow:** 只渲染阴影，不提供亮度照明。

**Sampling:** 采样模式，方式有两种，一种为 Adaptive QMC（自适应采样），另一种为 Constant QMC（常值采样）。



☐ Adaptive QMC（自适应采样）能够灵活地设

置场景中所需要的采样数，它将根据每个像素和相邻像素的亮度差异，自适应地产生数量不同的样本，并判断是否需要更多的采样计算，使像素差能满足极限值的标准。这种采样方式速度快，但是计算结果并不十分精确，效果也没有 Constant QMC 好。

☐ Constant QMC（常数采样）是对每一个像素都使用一个固定数量的样本做采样计算，它能提供分布比较均匀的采样阴影效果，结果更精确，但是速度较慢。

**Samples:** 采样值的大小定义了采样计算中，每个像素点上进行的最大采样计算次数。例如设置采样值为 3，那么采样次数就应该是  $3^2 = 9$  次。

**Soft Size:** 柔化尺寸, 用于控制临界位置的阴影扩散大小。值越小边缘部分的过渡越平滑, 但还需要结合较高的采样值设置, 才能获得更理想的阴影效果。如图 5-31 所示, 左图为  $\text{Soft} = 1$ ,  $\text{Sample} = 2$  时的渲染效果, 中图为  $\text{Soft} = 2$ ,  $\text{Sample} = 2$  的渲染效果, 而右图为  $\text{Soft} = 2$ ,  $\text{Sample} = 6$  的渲染效果。

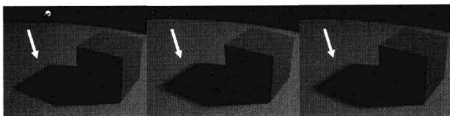


图 5-31 Soft 值须配合高 Sample 值才能得到理想的阴影效果

**Threshold:** 阈值, 配合 Adaptive QMC (自适应采样) 器, 用于确定采样器在哪些位置应使用最小比率, 哪些位置使用最大比率, 其值设置得越小, 渲染结果越精确。

## 5.2.2 缓冲阴影

### 5.2.2.1 Buffer Shadow (缓冲阴影) 的原理

Buffer Shadow (缓冲阴影) 是一种基于 Shadow Mapping (阴影映射) 算法下的阴影效果。Shadow Mapping (阴影映射) 算法是一种 Image-Space (图像空间) 算法, 它不需要了解物体的几何信息, 就可以产生正确的一面阴影。但是当将它应用于 Spot 之外的其他灯光时, 很容易出现边缘走样的问题。如图 5-32 左图所示, 为渲染右图时产生的缓冲阴影深度图。

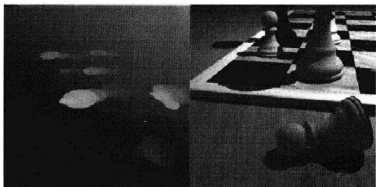


图 5-32 Buffer Shadow (缓冲阴影)

在进行渲染计算之前, 系统会建立一个缓冲区, 将离光源最近的物体表面与光源的距离值存储其中, 这个区又称为 Shadow Buffer (阴影缓冲) 区。接着在场景中, 系统将利用站在视角处沿光源法线看去所生成的深度图, 与阴影缓冲区做对比计算, 来检测场景中的体像素是否处于阴影中。

Shadow Buffer (阴影缓冲) 区是一个 2D 的深度图, 系统使用投影的方式将场景深度映射到 Shadow Buffer 中, 所以其运算的是深度值而非距离值。如图 5-33 所示, 系统会首先根据灯光映射出 Shadow Buffer, 当渲染  $P_a$  时,  $P_a$  与光源的深度等于 Shadow Buffer 中的值  $a$ , 所以  $P_a$  不在阴影中。当渲染  $P_b$  时, 其中  $b$  为  $L_b$  到光源的深度, 那么  $P_b$  与光源的深度将大于 Shadow Buffer 中的值  $b$ , 所以  $P_b$  就应该在阴影范围内。

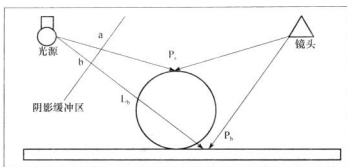


图 5-33 Shadow Mapping (阴影映射) 的算法原理示意图

### 5.2.2.2 Buffer Shadow (缓冲阴影) 的参数面板

还是以 Spot 灯光为例, 单击 Shadow 菜单即可找到 Buffer Shadow (缓冲阴影) 菜单, 如图 5-34 所示。

**Buffer Type:** 缓冲类型, 这里又提供了 4 种不同的缓冲计算模式, 包括 Classical (标准) 模式、Classic-Halfway (半缓冲) 模式、Irregular (不规则缓冲) 模式和 Deep (深度缓冲) 模式。

Classical 和 Classic-Halfway 功能类似, 后者为基于前者的改进算法, 现在为 Buffer Type 的默认选项。比起前者, 它优化了像素点间的 Z 深度值比较, 取平均值来平衡 Bias (偏移量), 从而避免出现阴影丢失的错误。使用这两种模式时必须合理设置 Bias (偏移量), 较大的 Bias (偏移量) 会忽视较小面上的阴影, 其中 Classic-Halfway 使用了算法来修正, 使得可以无需过于依赖 Bias (偏移) 取值来获得精确的阴影。但是基于这种算法本身在阴影重叠时可能出现的运算缺陷, 所以在有重复物体或有较小体积物体的场景中, 这两种模式都会出现一些小小的问题, 使阴影可能会变淡或者过于模糊。

首先选择 Classical (标准) 模式, 如图 5-35 左图所示, 是 Bias 为 1 的渲染结果, 可以看到在较大的 Bias 下, 基座位置较小面积的物体阴影被忽略掉了。右图是 Bias 为 0.2 的渲染结果, 可以看到降低 Bias 后得到了较好的效果。

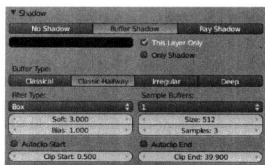


图 5-34 Buffer Shadow (缓冲阴影) 的参数面板

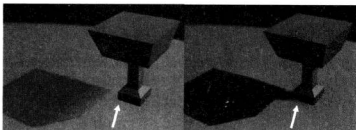


图 5-35 Classical (标准) 模式下的阴影效果

接着使用 Classic-Halfway, 如图 5-36 左图所示, 为 Bias 为 1 的渲染结果, 可以看到使用算法优化了的阴影效果, 可基座处的阴影依然由于较大的偏移设置而变得很淡, 不够真实。右图图 Bias 为 0.2 的渲染结果, 可以看到效果更好了。



图 5-36 使用改进算法和较小 Bias 值的渲染效果

**Bias:** 偏移量, 用于定义阴影起点与产生阴影的物体位置之间的位置偏移距离。当取值过大时会造成阴影偏离物体, 而过小会让物体与阴影产生干涉现象。解决的办法是适当调节 Bias 值, 或者使用具有算法优化的缓冲类型, 例如 Classic-Halfway 和 Irregular。如图 5-37 所示, 从左至右分别为 Bias 值为 1、0.1 和 0.001 的阴影效果。

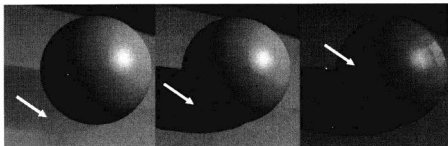


图 5-37 Bias (偏移量) 对阴影的影响效果

**Filter Type:** 过滤器类型。当渲染器在采样数据时, 每个采样点都会包含色彩和透明度等信息, 需要使用一定的方式来有效地选择其中的数据, 最简单的方式就是使用过滤器, 将处于非波峰范围的数据过滤掉。系统集成了三种不同类型的过滤器, 分别为 Box (矩形) 过滤、Tent (尖锐) 过滤和 Gauss (高斯) 过滤。如图 5-38 所示, 从左至右分别为它们的波形图。过滤器还需要配合合适的 Sample (采样值) 才能正常工作。当 Samples 为 1 的时候, 过滤器将默认为关闭而不会产生任何效果。

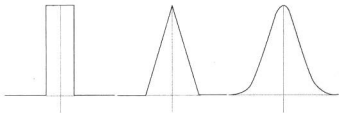


图 5-38 Box、Tent 和 Gauss 类型过滤器

- ❑ **Box** 矩形过滤器是系统默认的过滤器, 也是 Blender 最早使用的过滤器, 它能在较低的分辨率下获得较清晰的抗锯齿效果。在计算过程中它只考虑像素点上最密集的数据, 而会过滤掉周围分布较稀疏的采样数据, 因此常在具有尖锐边缘的物体上使用这种采样模式。如图 5-39 所示, 从左至右分别为 Samples (采样值) 是 1、4 和 9 时的阴影效果, 可以看到: 越高的采样率越能产生平滑的阴影边缘。

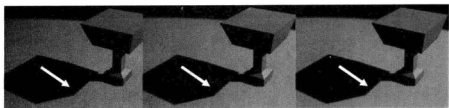


图 5-39 Box (矩形) 过滤器的平滑效果

- Tent (尖锐) 过滤使用了三角波来做采样过滤计算, 它能提供比较锐利的边缘效果。如图 5-40 所示, 从左至右分别为 Samples (采样值) 是 1、4 和 9 时的阴影效果。

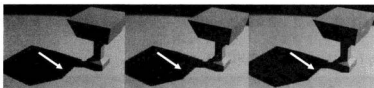


图 5-40 Tent (尖锐) 过滤器的平滑效果

- Gauss (高斯) 过滤采用了高斯算法, 它提供的是平滑的边缘效果, 因此十分适合高分辨率下的渲染输出。如图 5-41 所示, 从左至右分别为 Samples (采样值) 是 1、4 和 9 时的阴影效果。

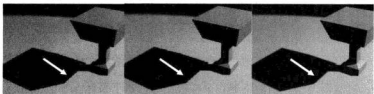


图 5-41 Gauss (高斯) 过滤器的平滑效果

**Soft:** 柔化, 柔化系数能定义阴影边界区域的过渡范围, 值越大临界处的过渡区域越大。如图 5-42 左图所示, 为 Soft 设置为 1 时的效果, 右图是值为 10 时的效果, 可见较高的柔化值下阴影的边缘柔和范围更大。

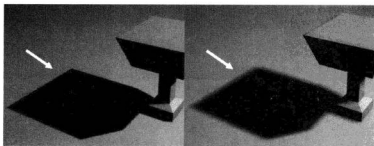


图 5-42 Soft 能柔和阴影边缘

不同于 Classic (经典) 模式采用的全景 Z 深度计算, Irregular (不规则) 缓冲模式改进了算法, 不再将每盏灯光的阴影都映射出来, 而只是映射可见物体部分的阴影, 并对这部分物体做采样。如图 5-43 所示为其控制面板, 由于它的自适应特性, 所以这里只保留了 Bias (偏移量) 设置。

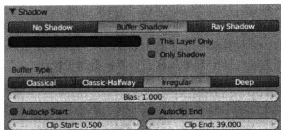


图 5-43 Irregular (不规则) 缓冲的参数面板

Irregular (不规则) 缓冲模式的优势在于, 它可以明显地提高渲染性能和阴影质量。系统不仅可以减少缓冲区的运算时间, 还能使用单独的线程来计算深度图, 并做预存储, 这对较大型的场景优化尤其明显。如图 5-44 左图所示, 需要使用优化的 Classic-Halfway 模式和较高的 Samples (采样值), 消耗 4.7M 内存, 花费 3 秒多才能得到一个满意结果。而如图 5-44 右图所示, 同样的效果, Irregular 只需要 4.68M 内存和 1 秒多的时间。

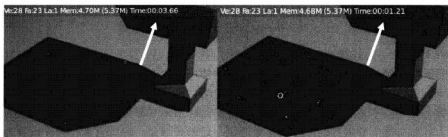


图 5-44 Irregular 的性能优化效果

Irregular 总能提供一个清晰的阴影边界, 而无需设置采样和柔化。这里 Bias (偏移量) 的影响效果和 Classical (经典) 模式下类似, 过大的调节偏移量设置会造成较小物体的阴影偏移, 丢失渲染精度。如图 5-45 所示, 左图为 Bias 为 1 的效果, 右图 Bias 值为 0.001, 可以看到箭头所示的阴影临界处的区别较大, 较小的 Bias 明显能得到更精确的阴影效果。

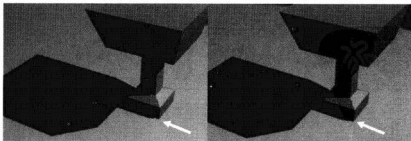


图 5-45 Bias 对临界阴影的影响

但是 Irregular 也存在一些缺陷, 由于这种算法是基于 Scanline (扫描线型) 的渲染原理, 当场景使用了光线追踪的镜面反射、光线追踪的透明效果和体积光效果时, Irregular 产生的阴影就容易出现错误, 甚至不可见。

**Deep:** 深度缓冲模式, 它可对 Shadow Map 中的深度分布和透明阴影做预过滤计算, 制作出真实的阴影效果, 常用于强化毛发等较小体积的物体渲染、体积云和烟雾以及快速运动中的物

体阴影。但是基于其运算特性，Deep 模式会消耗掉大量的内存空间和存储空间做预处理计算，因此在 2k 级别以下的渲染中，应控制缓冲值在 1024 以内，以获得平衡的渲染性能。如图 5-46 左图所示为光线追踪下的阴影计算效果，如图 5-46 右图所示为使用了 Deep（深度）缓冲计算的毛发渲染效果，可以看到毛发的阴影效果和真实感明显得到了优化。

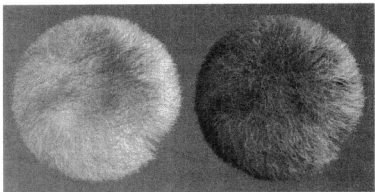


图 5-46 Deep（深度）缓冲模式渲染效果

**Sample Buffers:** 采样缓冲，用于定义系统对阴影做抗锯齿运算时需要的缓冲次数。当计算较小物体阴影时，例如绳索和毛发，即使在运算时使用了较高的阴影分辨率，也不一定能达到预期效果。所以这时可以使用多次缓冲采样计算的方法，来优化阴影的效果，当值为 1 时功能将默认为关闭状态。要注意一点，当在使用一个较高的采样值来平滑抗锯齿效果时，系统为了产生光滑的效果，需要在每个像素点的采样阴影上产生一定的偏移量，假设当采样缓冲设置为 4 时，多次的缓冲计算会使花费的时间和消耗的内存增加到原来的 4 倍。

**Size:** 缓冲大小定义了阴影分辨率，Size 值越高，阴影的解析度越高，效果也越精确。例如，当 Size 值为 512 时，阴影单位采样面上的分辨率为  $512 \times 512$ 。如图 5-47 所示，从左至右分别为 Size 值是 512、1024 和 4096 时的阴影渲染效果图。

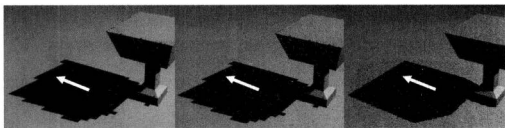


图 5-47 Size 值对阴影解析度的影响

## 5.3 灯光类型

Blender 内置了 5 种不同类型的灯光，每一种都有特殊的作用和不同的功能。可以在视图窗口中单击 Shift + A 来添加灯光，它们在选择器模式和线框模式下的显示效果都是透明的。

### 5.3.1 点灯光

Point Light（点灯光）是一种全方位发散的灯光，在视图中显示为一个带圆心的圆圈，如图 5-48 所示。Point（点灯光）和 XY 平面的连线表示了它的相位，如果灯光在 Z 轴的正方向，



那么水平面将向下牵引着 Point（点灯光），其距离值使用了一条虚线来表示。

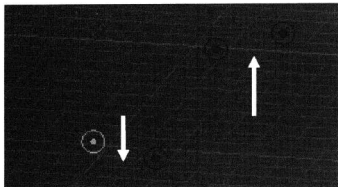


图 5-48 Point Light（点灯光）

如图 5-49 所示，为 Point（点灯光）的面板。点灯光只能产生 Ray-Tracing 阴影，具体参数可以参考前面的简介。

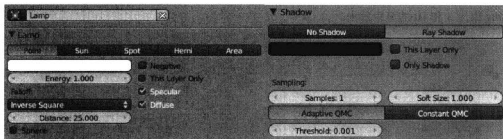


图 5-49 Point（点灯光）的参数面板

点灯光的光线效果发散性强，灯光对物体的照明力度取决于两者之间的距离，物体表面离灯光越远，接收到的光照强度越弱。如图 5-50 所示，可以看到，离灯光距离最远的物体表面最暗。

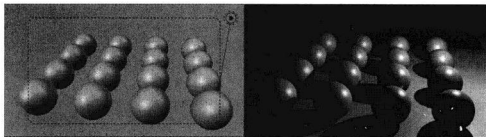


图 5-50 Point Light（点灯光）的照明效果

### 5.3.2 日光

Sun Light（日光）提供了太阳光的模拟光源，它可以按照一定的方向辐射到场景中，如图 5-51 左图所示为 Sun（日光）的参数面板。常规设置和其他灯光类似，可使用 Ray-Tracing 阴影，它没有衰减选项，默认使用了 Constant（常值）衰减模式。日光面板中的天空和大气选项比较特殊，如图 5-51 右图所示。

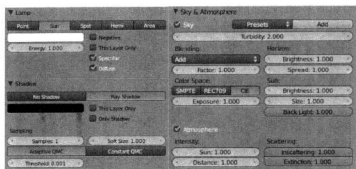


图 5-51 Sun Light (日光) 的参数面板

日光不仅需要模拟太阳的光线特性,还要模拟天空和大气的一些特殊效果。例如当正午时刻时,天空应该是天蓝色,而当太阳处于西落的时候,天空应呈现出一些暗蓝色或者紫色。由于太阳高度决定了场景所在的时间,所以 Sun 的角度设置很重要。

**Turbidity:** 浊度, 较低的设置可获得干净淡蓝的天空效果, 而 Turbidity 值越高大气会显得越混沌, 天空呈现出暗红色, 太阳周围也将产生光晕效果。如图 5-52 所示, 从左至右分别为 Turbidity 值为 2 和 10 的渲染效果。

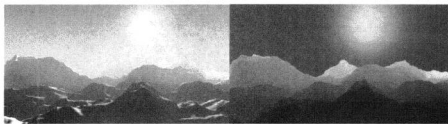


图 5-52 不同 Turbidity (浊度) 的效果

**Blending:** 混合模式, 用于混合环境设置中的天空和太阳, 其中 Factor 为混合值。

**Color Space:** 色域管理, 用于转换特定的色域类型, 以满足不同媒体介质的兼容性。例如在 LCD 上查看 RAW 格式的文件, 对比在 CRT 上查看同一文件的 JPEG 格式, 就会发现颜色略有不同。设置中的 SMPTE 常用于电视系统, REC709 则是 HDTV 的色域标准, 而 CIE 是显示器的 RGB 输出标准。

**Exposure:** 曝光系数, 用于控制色彩的曝光强度。

**Horizon Brightness:** 控制水平方向的色彩亮度, 当值为 0 时水平方向无颜色亮度。如图 5-53 所示, 从左至右分别为 Brightness 值为 0 和 1.5 的效果。

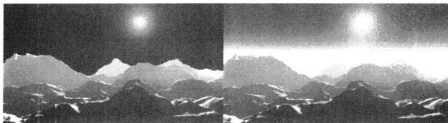


图 5-53 Horizon Brightness (水平方向的色彩亮度) 影响效果

**Horizon Spread:** 水平平铺参数, 用于控制水平方向上的光线拉伸效果。如图 5-54 所示, 从左至右分别为 Spread 值是 0.5 和 5 的效果。



图 5-54 Horizon Spread（水平平铺）的效果

**Sun Brightness:** 控制太阳光的亮度，值越低太阳光越弱。如图 5-55 所示，从左至右分别为 Brightness 值是 0.2 和 1 的效果。



图 5-55 太阳光的 Brightness（亮度）效果

**Sun Size:** 设置太阳的尺寸，较低的值代表较大的太阳尺寸。如图 5-56 所示，从左至右分别为 Size 值是 2 和 10 的效果。



图 5-56 Sun Size（太阳的尺寸）效果

**Back Light:** 太阳周围的光线效果，值越高太阳的边缘越亮。如图 5-57 所示，从左至右分别为 Back Light 值是 -1 和 1 的效果。

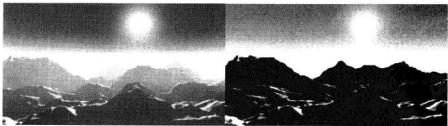


图 5-57 Back Light（太阳周围）的光线效果

**Atmosphere:** 大气效果，单击后可启用大气模拟。

**Sun Intensity:** 太阳光的强度，当设置较高的强度值时，远处的物体上会出现较多的蓝色光

照效果。如图 5-58 所示，从左至右分别是 Sun Intensity 值为 2 和 10 的效果。



图 5-58 Sun Intensity（太阳光的强度）效果

**Distance:** 光程，这里可以将 Blender 单位转换为其他单位，以方便大气效果对场景的影响。如图 5-59 所示，从左至右分别为 Distance 值是 1 和 4 的效果。

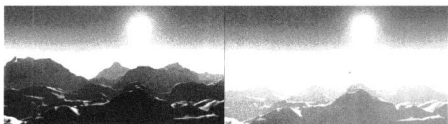


图 5-59 Distance（光程）的影响效果

**Inscattering:** 内散射，用于降低大气中摄像机和物体之间的散射光数量，常把它设置为 1。如图 5-60 所示，从左至右分别为 Inscattering 值为 0.1 和 1 的效果。



图 5-60 Inscattering（内散射）的影响效果

**Extinction:** 消散，用于调节场景中物体对光的消散程度，常设置为 1。如图 5-61 所示，从左至右分别是 Extinction 值为 0.1 和 1 的效果。



图 5-61 Extinction（消散）的效果

### 5.3.3 聚光灯

Spot（聚光灯），又叫射灯，它能按照指定的方向，向一个锥形区域的范围内投射光线。与其他灯光比较，Spot 灯光是功能最复杂，也是应用最广泛的一种灯光物体。它不仅能产生 Ray-Tracing（光线追踪）阴影，还能产生 Buffer Shadow，同时还是唯一支持体积光的光源。如图 5-62 所示，为 Spot（聚光灯）的参数面板。

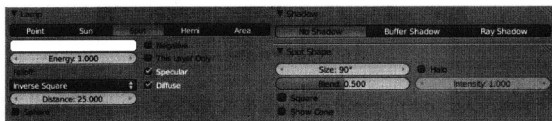


图 5-62 Spot（聚光灯）的参数面板

Spot（聚光灯）的 Distance（光程）等参数决定了它在 3D 视图中的形态，如图 5-63 所示，为不同 Distance（光程）参数时的灯光形状。

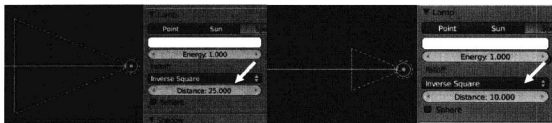


图 5-63 不同 Distance（光程）距离下的灯光形状

在 Spot 灯光的 Buffer Shadow（缓冲阴影）属性中，可以设置 Spot 灯光特有的属性 Clip（间断）参数，其中 Clip Start 和 Clip End 分别定义了产生 Buffer Shadow（缓冲阴影）的范围。如图 5-64 所示，形象地标识了 Spot 灯光的结构参数，当单击 Auto Clip 时，灯光将根据光程距离和衰减模式来自动计算 Clip 距离。

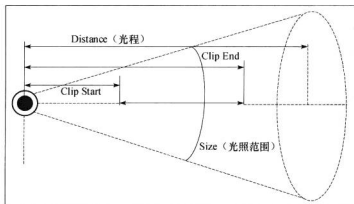


图 5-64 Spot 灯光的结构参数示意图

添加一个简单的场景，如图 5-65 左图所示。接着开启 AutoClip 功能，灯光将根据光程自动调整阴影的范围，最后的渲染效果如图 5-65 右图所示。注意：灯光只能照明锥形范围内的区域。

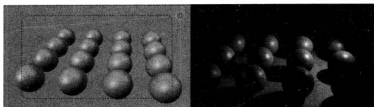


图 5-65 Spot 聚光灯的照明效果

### 5.3.4 半球灯

Hemi（半球灯）提供了一个 180 度的半球形光照模型，如图 5-66 左图所示为其原理图，它提供了一个全方位的包围型光源。Hemi 在视图中的效果，就像在一个点光源上添加了 4 条弧线，指向了光照的方向，如图 5-66 右图所示。

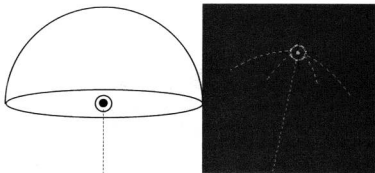


图 5-66 Hemi（半球灯）原理和形态图

相比于其他灯光，Hemi 的面板相对简单了很多，它只提供了基本的光照颜色和光强设置，同时它默认使用 Constant（常值）衰减模式，取消了光程距离等选项，因此 Hemi 的光照效果相对来说比较单一。如图 5-67 所示，为 Hemi（半球灯）的参数面板。

如图 5-68 左图所示，在场景中添加一个 Hemi 灯光，单击渲染后，结果如图 5-68 右图所示。可以看到所有的球体都因为 Hemi 的包围照明特性，得到了十分全面的照明。由于 Hemi 是唯一不会产生阴影的光源，所以可以看到右图中物体的下方都没有任何阴影，因此 Hemi 灯常特别适合制作卡通效果。

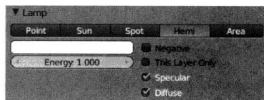


图 5-67 Hemi（半球灯）的参数面板

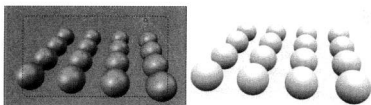


图 5-68 Hemi（半球灯）的照明效果

### 5.3.5 面光源

Area Light（面光源）能模拟出一种由表面发光的光照效果，可以提供十分柔和的阴影。例如生活中的电视屏幕或者广场大屏幕，就是一种典型的面光源。如图 5-69 所示，为面光源的参数面板。其中需要注意的是：面光源对 Distance（光程距离）参数十分敏感，一般来说在光程范围内的所有物体都可能会出现曝光过度的现象，所以常将物体位置放置在远离光源的光程末端。

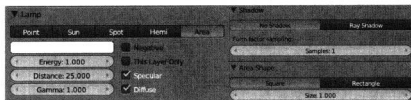


图 5-69 Area Light（面光源）参数面板

同时还要注意的：Area Light（面光源）没有衰减选项，它默认使用 Inverse Square（反向二次曲线）的衰减模式，同时使用了 Gamma（伽玛值）来自动调节衰减力度。当 Gamma 值越大时，照明的对比度越强，同时光线衰减得也越快。

**Area Shape：**面外形，用于定义发光面的形状，Square（正方形）选项下 Size 尺寸用于设置发光面的矩形大小，也可以在 Rectangle（矩形）选项中，分别定义 SizeX 和 SizeY 的大小。

和 Spot 一样，灯光的参数决定了它在 3D 视图中的形态。如图 5-70 所示，面光源的法线方向决定了光照方向，矩形的大小决定了光照的范围。

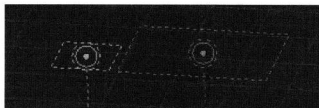


图 5-70 Area Light（面光源）形态

**Samples：**Area Light（面光源）开启 Ray Shadow（光线阴影）中的采样值，用于控制面光源产生的阴影效果。当提高采样值参数时，面光源会根据采样数对发光面做细分操作。如图 5-71 左图所示，当 Samples 为 1 时，面光源的发光面就相当于一个点光源。如果改变设置 Samples（采样值）为 3，那么发光面将被细分为一个近似  $3 \times 3$  的网格，网格中的每个点都将成为一个点光源。由于每个光源都将产生一个阴影，所以当 Samples 为 3 时，如图 5-71 右图所示，球体后面产生了 9 个阴影。

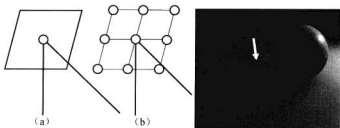


图 5-71 面光源的采样原理及效果

那么由于细分后而形成了多点照明，面光源下的物体就会因此产生多个阴影，多阴影的重叠将产生柔和的阴影效果。如图 5-72 左图所示，为 Samples 为 1 的渲染结果，可以看到阴影边缘十分清晰，当把 Samples 增加到 6 时，如图 5-72 右图所示，可以看到阴影十分柔和，但是渲染时间却增加了近两倍。

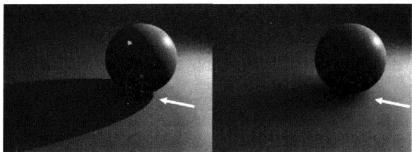


图 5-72 面光源的高采样值阴影效果

在阴影设置中，面光源还有一项独有的采样方式，Constant Jittered（连续抖动）采样。如图 5-73 所示，为 Constant Jittered（连续抖动）采样的参数面板。

**Umbra:** 使系统能自动强调阴影的最暗部区域。

**Dither:** 启用阴影抖动采样，可用于平滑阴影间相互重叠的部分。

**Jitter:** 使用噪点来平滑阴影边缘。在低采样值的情况下，可以适当地调节 Dither（抖动）和 Jitter（噪点）来优化阴影力度，从而达到一个快速柔和阴影的结果。如图 5-74 左图所示为优化前的效果，右图为激活 Dither 和 Jitter 的效果。

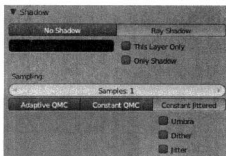


图 5-73 Constant Jittered（连续抖动）采样的参数面板

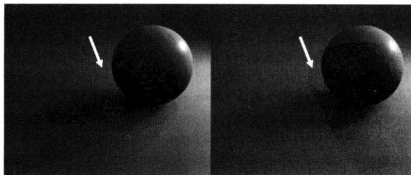


图 5-74 Dither（抖动）和 Jitter（噪点）对阴影的优化

## 5.4 环境属性

仅仅灯光的自带效果还不能为场景提供真实的光模拟环境，Blender 还提供了一些辅助环境增强的功能。例如 Ambient Occlusion（环境光吸收）、Environment Lighting（环境光照）、Indirect



Lighting (间接光照) 和 Mist/Stars (迷雾与星辰) 等。

### 5.4.1 环境与背景

在 World (世界) 属性面板中, 可以为环境配置背景光颜色和模式。如图 5-75 所示, 为 World 世界属性面板。其中可以分别调节 Horizon (地平线)、Zenith (天空顶点) 和 Ambient (环境的颜色)。

**Paper Sky:** 平铺天空, 单击后地平线的颜色将始终保持在摄像机的中心。

**Blend Sky:** 混合天空, 单击后背景将使用混合的方式来过渡地平线和天空顶点间的颜色。

**Real Sky:** 真实天空, 单击后系统将在渲染时根据摄像机的角度产生真实的地平线效果。

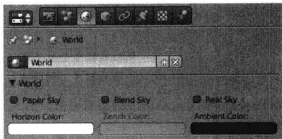


图 5-75 World (世界) 属性面板

### 5.4.2 环境光吸收

在真实的大自然中, 光从太阳照射到地面将经过无数次的反射和折射, 因此人们看到地面的任何地方都是清晰的。在渲染的时候, 为了实现真实的场景效果, 可以在渲染器中使用 Global Illumination (全局光照) 的原理, 实现一种可欺骗视觉的物理假象。使光从光源发射出来后, 当碰到障碍物时就会产生反射和折射, 在经过无数次的反射和折射后, 物体表面和角落才会有光感, 看上去像真实的自然光照明效果一样。由于全局光照是采用 Photon Map (光子贴图) 来实现的, 所以全局光照的方法十分消耗内存和渲染时间。

Ambient Occlusion 简称 AO, 是一种可以用来模拟 GI 全局照明结果的技术, 最早是在 Siggraph 2002 年会上由 ILM (Industrial Light & Magic, 工业光魔) 的技术主管 Hayden Landis 提出, Blender 是较早引入 AO 这项技术的软件之一。AO 可以模拟光线的能量传递, 以独特的方式吸收未被阻挡的光线和被阻挡的光线, 并利用两者间的能量关系产生出阴影效果, 从而模拟出物理照明的效果。AO 可以为渲染贡献非常精确和平滑的阴影, 就好像是 GI 全局照明的结果。同时它改善了内置渲染漏光、飘虚和阴影不实的缺点, 以及场景中的缝隙、褶皱与墙角、角线等细小物体处表现不清晰的问题。如图 5-76 所示, 左图为未添加 AO 的渲染结果, 中图为 AO 的运算阴影图, 右图为将 AO 叠加在正常渲染上的效果。可以明显看到: 使用了 AO 之后, 渲染的画面明暗对比被加强了, 同时表面的阴影也增强了空间上的层次感和真实感, 进一步丰富了细节。

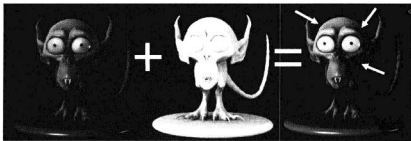


图 5-76 使用 AO 优化后的渲染结果

#### 5.4.2.1 AO 运算原理

AO 的计算不需要任何灯光照明的支持, 它将主要收集光线在物体上的折射、衍射、散射和吸收数据, 使渲染在受影响的位置上产生适当的叠加数据, 来改善阴影效果以显示更好的图像

细节。特别是在由于场景物体较多阻挡光线传播而导致间接照明不足时，它可以更加明显地表现周边所有环境的阴影。AO 的计算过程是首先在几何体表面任意一点的上方，使用预定半径的半球来探测该点的外部区域，从而决定光线是否在该几何体表面上被其他几何体阻挡，或者被吸收。一旦各点的吸收幅度被确定，则几何体表面将根据得到光照强度的能力形成一张灰度级的映射图，映射图中的明暗区域就可以按比例来辅助渲染器，调节物体表面对环境光吸收的强弱程度，以此形成环境光对该几何体表面的照明影响。如图 5-77 所示，地面上的 A 点由于探测光线的可吸收路径较长，所以在视觉光线下看上去就会更亮，而 B 点由于探测光线在较短的距离内就被地面物体给阻止并吸收掉了，所以这一点在视觉光线下看上去就会更暗。

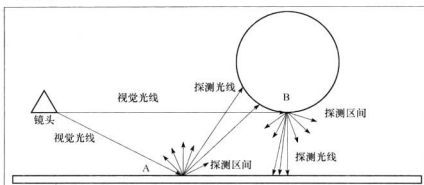


图 5-77 AO 的运算原理示意图

由于 AO 并没有直接计算灯光的光线，而是计算环境的属性，所以即使没有灯光，AO 也能产生一定的效果。如图 5-78 所示，为 AO 的参数控制面板，可以在环境类属性中找到。

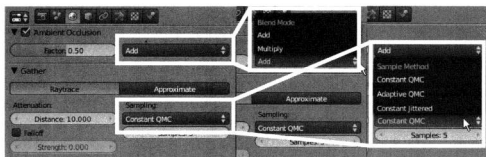


图 5-78 AO 的参数控制面板

**Factor:** 用于控制 AO 的运算强度，值越高，运算的时间越长，效果越好。

**Blend Mode:** 混合模式，可选择 Add（添加）和 Multiply（叠加）两种模式。

#### 5.4.2.2 Raytrace（光线追踪）模式

Raytrace（光线追踪）的效果最精确，但是会产生很多噪点。如果要消除噪点，可能需要付出更多的渲染时间。Raytrace（光线追踪）一共提供了三种采样模式。

**Constant QMC:** 连续性采样，能提供全面而平均的分布式采样计算效果。如图 5-79 所示，为 Constant QMC（连续性采样）的参数面板。

**Adaptive QMC:** 自适应采样，这是基于 Constant QMC 的一种改进采样模式。如图 5-80 所示，为 Adaptive QMC

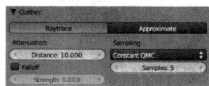


图 5-79 Constant QMC（连续性采样）的参数面板

(自适应采样)的参数面板。其中 Threshold (采样偏移), 用于定义自适应采样计算时的允许偏移量范围。Adapt To Speed 可结合 Speed Vector (速度矢量) 通道, 来优化物体在运动中的 AO 像素采样点尺寸, 因此在渲染时需要开启 Vec 速度矢量通道。

**Constant Jittered:** 常值抖动采样, 它使用 Bias 偏移量来控制采样中对物体边缘的平滑效果。如图 5-81 所示, 为常值抖动采样的参数面板。对于一个已被 Smooth (平滑) 工具修改过尖锐面的物体, 如果使用较小的 Bias 来计算 AO, 面的边缘细节还是可以被显示出来。

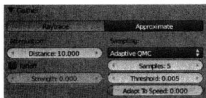


图 5-80 Adaptive QMC (自适应采样) 的参数面板

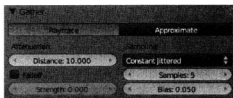


图 5-81 Constant Jittered (常值抖动) 采样的参数面板

**Samples:** 采样值, 是采集数据和运算的次数, 也是 AO 的基本参数。较高的采样值能获得更好的采样结果, 同时还能减小噪点。如图 5-82 所示, 从左至右分别为采样值为 1、6 和 12 的渲染效果, 可以观察物体表面上的噪点差异。

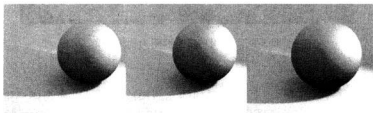


图 5-82 较高的 Samples (采样值) 可降低噪点

**Distance:** 探测幅度, 如果提高这个值, 渲染时系统会花费更多的时间来计算远距离, 采样点探测幅度内的物体遮挡和灰度吸收值。为了减少渲染时间, 建议采用较小的 Max Distance 值设置。

#### 5.4.2.3 Approximate (近似采样) 模式

Approximate (近似采样) 模式的效果更平滑, 而且渲染时间更快, 但是精度要稍微差一点。如图 5-83 所示, 为 Approximate (近似采样) 模式的参数面板。

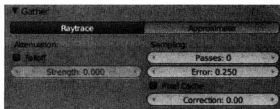


图 5-83 Approximate (近似采样) 模式的参数面板

**Falloff:** 衰减选项, 激活后可调节 Strength 衰减力度, 值越高亮度向阴影的过渡衰减得越快。

**Passes:** 采样通道, 较高的采样通道数能获得更好的效果, 但是会消耗成倍的计算时间。

**Error:** 容差, 值越小采样精度越高, 默认为 0.25。

**Pixel Cache:** 开启像素缓存功能, 能提高渲染速度, 但是可能会损失一些小细节。

**Correction:** 校正值, 值越大对曝光运算结果的校正力度也越强。

### 5.4.3 环境光照

如图 5-84 所示，为 Environment Lighting（环境光照）的参数面板。

Environment Lighting（环境光照）功能，用于开启环境对场景的光照能力，这里提供了 3 种照明模式：White（白色）、Sky Color（天空色彩）和 Sky Texture（天空材质）。开启对

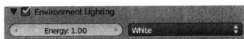


图 5-84 Environment Lighting（环境光照）的参数面板

应的模式后，系统将使用对应的环境色彩对场景做辅助全局照明，使一些没有受到直接光照的区域得到环境的补光，常用于动画的制作中。如图 5-85 左图所示，为未使用环境光照的一块石头的渲染效果，而右图图为使用 White 模式、Energy（光照强度）为 0.4 的渲染效果，可以看到物体得到了全方位的补光照明。

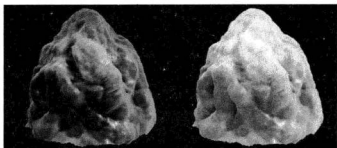


图 5-85 使用环境光照明为物体补光

### 5.4.4 间接光照

Indirect Lighting（间接光照），是一种辅助的光照模式，它可以使物体产生自发光的效果，用于模拟灯泡、霓虹灯等一切可自发光的物体照明。要开启间接光照，需要将 Gather 模式设置为 Approximate，如图 5-86 所示，为 Indirect Lighting（间接光照）的参数面板。



图 5-86 Indirect Lighting（间接光照）的参数面板

**Factor:** 用于设置物体的间接光照强度，值越高，间接光的照明力度越强。

**Bounces:** 反弹次数，用于设置参与 Approximate（近似采样）计算的光线反弹次数。如图 5-87 所示，为使用 3 个物体做间接光照光源的渲染效果，其中发光的物体应在其材质选项中提升 Emit（发光）属性的数值设置，并开启 Shading（着色）属性的 Cubic Interpolation（矩形插值）计算选项。

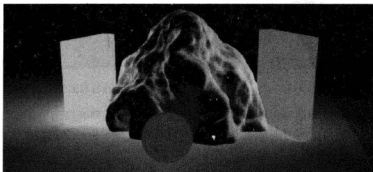


图 5-87 Indirect Lighting（间接光照）的渲染效果

## 5.4.5 迷雾与星辰

### 5.4.5.1 Mist (迷雾)

World (世界) 属性面板还提供了一个自动模拟迷雾和星辰的选项。如图 5-88 所示, 为 Mist (迷雾) 的参数面板。

**Intensity:** 迷雾的整体强度, 值越大雾越浓。

**Start:** 与摄像机产生迷雾渐进效果的起始距离。

**Depth:** 迷雾深度, 所有与摄像机的距离大于 Start + Depth 数值的物体, 都将被完全隐藏在雾中。

**Height:** 设置迷雾的海拔高度。

**Falloff:** 迷雾衰减率。

当要查看迷雾效果时, 必须激活摄像机的迷雾显示功能。如图 5-89 左图所示, 单击摄像机后进入其控制面板, 激活 Mist 选项即可。如图 5-89 右图所示, 为一个简单的迷雾效果图。



图 5-88 Mist (迷雾) 的参数面板

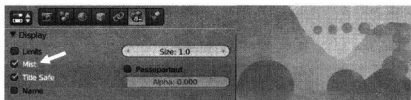


图 5-89 激活摄像机的迷雾显示渲染迷雾效果

### 5.4.5.2 Stars (星辰)

如图 5-90 所示, 为 Stars (星辰) 的参数面板。

**Size:** 星辰尺寸。

**Colors:** 产生随机的星辰颜色。

**Min. Dist:** 星辰与摄像机之间的最小距离值。

**Separation:** 定义星辰之间的距离值。

如图 5-91 所示, 使用星辰功能制作太空等场景时, 可以很轻松地添加一些外星背景。



图 5-90 Stars (星辰) 的参数面板

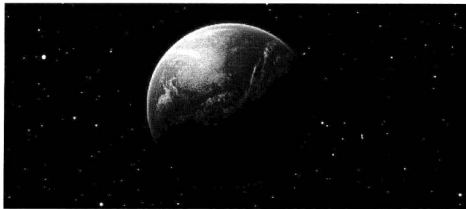


图 5-91 星辰效果图

### 5.4.6 HDRI 照明

除了色彩和灯光辅助功能，Blender 还可以为环境添加贴图纹理，例如可以使用 HDRI 文件来照亮整个场景。

HDRI (High-Dynamic Range Image, 高动态范围图像) 是一种特殊的文件格式，它记录了大量的亮度范围数据，具有足够的光照信息和能力，常用于影视特效和游戏实时渲染中。将 HDRI 贴到环境的照明通道中，通过读取图片上的亮度细节，就可以产生真实的反射和折射光线效果。如图 5-92 所示，首先单击 World (世界) 属性，然后转入材质选项，单击 Add 添加一个新的贴图文件，选择纹理类型为 Image (图片)。

接下来单击 Open 按钮，选择要添加的 HDRI 文件，如图 5-93 左图所示。转入 Influence (影响) 范围选项，激活 Zenith Up 地平面上和 Zenith Down 地平面向下，设置如图 5-93 右图所示。



图 5-92 为环境添加贴图



图 5-93 选择 HDRI 文件并设置影响通道

接着建立一个简单的场景，不添加任何灯光，如图 5-94 左图所示。单击 F12 渲染，就可以看到物体表面得到了 HDRI 图上的所有照明信息，效果十分真实自然，如图 5-94 右图所示。

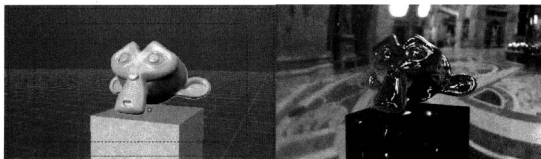


图 5-94 HDRI 的渲染效果图

## 5.5 色彩管理和线性工作流

Blender 采用了 Linear Workflow (线性工作流) 的模式，来实现色彩空间的输入与输出管理。由于计算机系统和人眼对色彩的识别能力存在差别，通过计算机渲染出来的画面很难实现人肉

眼对真实世界的体验与重现。首先，世界中颜色是无穷连续变化的，这是一个线性值，但是人眼对色彩和亮度的感知却是非线性的，它是一种指数型的分布。当真实世界中的光线到达人眼时，人的视觉神经会对人眼接收到的光线做一次非线性“校正”，使光线和人们的视觉系统产生色彩匹配。这种“校正”就是将真实世界中的色彩，调整为人们头脑中自己的色彩系统，让眼睛看到的東西更接近真实色彩。

接着来看计算机世界，它在系统内部是可以实现一个线性色彩空间的，用来对数据进行线性化处理。但是问题就是，如何通过人眼视觉来确定计算机处理的颜色是所需要的颜色呢？而且对于显示器，由于其亮度值将随电压的变化产生波动，所以视觉上本身就已经出现了失真；对于显示的色彩，不同的操作系统对于色彩的管理也存在偏差。这样多种的原因就造成了各种色彩空间无法产生一个统一的标准，用于保证计算的可重复性和色彩的准确性。

解决的办法就是使用一个 Linear Workflow（线性工作流）的方式，将所有的用户输入数据从可视的非线性空间，校正为线性数据，然后在系统的线性空间中执行计算，最后再转换为我们肉眼可识别的指数非线性色彩，并在显示器上执行查看和保存操作。这样的流程就可以保证运算的颜色与物理的标准统一，以及色彩在跨平台上的统一。当然，对于硬件层面来讲，我们还需要手动校准显示器的色彩显示，使其 Gamma（伽玛值）能与空间转换时应用的 Gamma 系数相同，减少视觉误差。

在 Blender 中，当在渲染面板中开启了 Color Management（色彩管理）选项，系统将自动激活线性工作流，所有的输入数据（包括着色器、贴图、灯光等）都将被提前转换为线性化标准。如图 5-95 所示，为线性工作流中对色彩的转换流程。关于线性化合成，将在后面的第 10 章结点应用中做详细讨论。

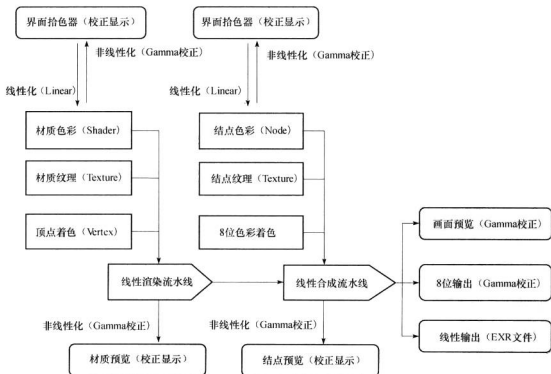


图 5-95 线性工作流中对色彩的转换流程

## 第 6 章

# 材质与贴图

物体在渲染前需要为其添加一个 Material（材质），使其可以在渲染中模拟现实中的材料和表面质感。物体的材质属性包括了 Shader（着色器）和 Textures（纹理）贴图。

## 6.1 着色器

Shader（着色器）是渲染器的一部分，负责计算目标物体的颜色和材质。我们制作的绝大多数物体都是由网格构成，它们都必须经过一定的着色处理，才不会以线框的方式，显示在视图窗口或渲染结果中。视图中的着色采用了 OpenGL 的计算方式而渲染中的着色则采用了渲染器的着色流水线。本章将着重讲解渲染中着色器的应用。首先，先简单了解一下物体材质效果在渲染中的计算原理。

### 6.1.1 材质的渲染原理

渲染器的计算原理其实和真实世界中的人眼成像原理类似，它根据灯光和环境等参数来计算物体的材质等相关模拟数据，并使用映射的方式，将场景投影到平面视图上形成渲染图像。渲染前系统会先分析场景中的有效光线和物体之间的关系，如图 6-1 所示，为一个场景中灯光、物体和镜头的位置关系图。渲染就像是从镜头上的视觉点出发，向可渲染物体的表面引出一条光线的反向路径，它与物体表面的夹角就是反射角。在该点上可以找到产生这一点上反射光线的灯光，灯光与这点之间的连线与表面之间的夹角就是光线的入射角。

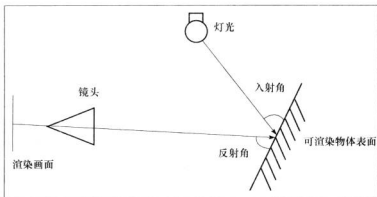


图 6-1 渲染的成像基本原理图

当光线发射到物体表面上时，由于入射角和反射角的角度不同，会产生两种基本的光学现



象：漫反射（散射）和镜面反射，这两种现象的可视效果就构成了我们对物体的主要材质官感。

对于透明的物体，一部分光会穿过透明体，产生透射光，而只有部分反射光和透射光可见，这时就需要配合反射率和折射率参数，来调节物体表面接收的光线强度。




因此，渲染器对物体的 Shading（着色）计算过程，就是将物体的基本颜色和接受光照强度等物理因素考虑在内，根据材质上类似的属性参数，模拟光线与物体之间光学关系的过程。

### 6.1.2 材质选项

当单击选择一个物体后，我们就可以在功能面板中找到其 Material（材质）选项，单击 New 即可为当前物体新建一个材质。如图 6-2 所示，为 Material（材质）的参数面板。



图 6-2 Material（材质）的参数面板

一个物体可拥有多个 Material（材质），单击  按钮可继续为当前物体添加新的 Material（材质），同理使用  按钮可删除现有的 Material（材质）。一个 Material（材质）可以被多个物体同时调用，单击  按钮在下拉菜单中选择现成的 Material（材质），并在 Data 选项中修改 Object（物体）与 Material（材质）的数据链接方式即可。

材质面板一共提供了 4 种材质的渲染模式，分别为 Surface（表面）材质、Wire（线框）材质、Volume（体积）材质和 Halo（光晕）材质。

#### 6.1.2.1 Preview（预览）面板

在 Preview（预览）面板中，一共有 6 种物体模式供用户选择来测试材质的效果。如图 6-3 所示，分别为 Flat（平面）、Sphere（球体）、Cube（立方体）、Monkey（大圣）、Hair（毛发）和 Large Sphere with Sky（带场景的球体物体）预览效果。预览面板是实时观察材质调节效果的地方，针对不同类型的渲染模式，应选择合适的预览物体，才能得到正确的测试效果。

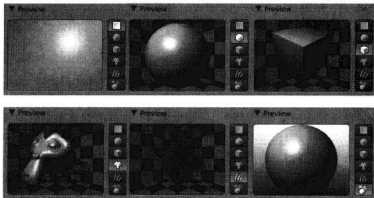


图 6-3 材质的 6 种 Preview（预览）模式

### 6.1.2.2 Shading (着色) 选项面板

这里主要包括了与着色器相关的通用属性,如图 6-4 所示,为 Shading (着色) 选项的参数面板。

**Emit:** 自发光属性,默认为 0 时,物体不会产生自发光的效果。

**Ambient:** 影响材质的环境光数量,默认为 1 时,物体可 100% 接受场景中光线的影响。

**Translucency:** 开启背部漫反射的影响效果,默认为 0。

**Shadeless:** 取消着色效果,渲染时物体表面将不会产生任何光学现象。

**Tangent Shading:** 使用切向着色而非法向着色,可制作出各向异性的效果,例如金属锅具上的拉丝纹理等。如图 6-5 所示,为开启 Tangent Shading (切向着色) 的一些渲染效果。

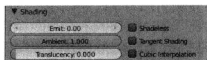


图 6-4 Shading (着色) 选项的参数面板

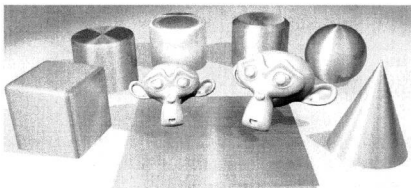


图 6-5 使用 Tangent Shading (切向着色) 制作的各向异性效果

**Cubic Interpolation:** 使用矩形插值的方式来计算物体的明暗过渡效果。

### 6.1.2.3 Strand (线股) 选项面板

当使用粒子来制作毛发等物体时,需要开启材质的 Strand (线股) 选项,来修改粒子在渲染时的显示效果。如图 6-6 所示,为 Strand (线股) 选项的参数面板。

**Size:** 尺寸,使用 Root (根部) 尺寸和 Tip (顶端) 尺寸参数,分别控制发根和发尖的形态。

**Blender Units:** 使用 Blender 单位来计量粒子的着色尺寸。

**Tangent Shading:** 根据毛发的方向,实现切线着色的效果。

**Shape:** 外形指数,值越低,毛发越纤细。

**Width Fade:** 使毛发在渲染时产生横向的淡出效果,这样看起来更柔软,当值为 1 时,透明淡出为线性效果。

**Surface Diffuse:** 启用网格表面的漫反射叠加计算,使系统可以利用物体表面的法线,将网格上的光照阴影效果应用至毛发的漫反射效果上,其中 Distance (距离) 值为允许使用的法线长度。这样制作出来的毛发照明色彩和亮度会更接近网格上的照明效果,特别适合于制作较短的毛发。如图 6-7 左图所示,为未启用 Surface Diffuse 的毛发渲染效果,中图为该网格表面的光照

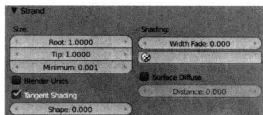


图 6-6 Strand (线股) 选项的参数面板

阴影效果。当开启 Surface Diffuse 选项后，系统会将两者的效果叠加起来，如图 6-7 右图所示。

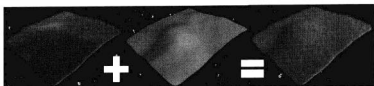


图 6-7 Surface Diffuse（网格表面的漫反射）计算效果

#### 6.1.2.4 Options（选项）面板

如图 6-8 所示，为材质设置的 Option（选项）的参数面板。

**Traceable:** 激活使用该材质物体的光线追踪计算。

**Full Oversampling:** 在渲染时开启材质的抗锯齿采样。

**Sky:** 使用天空的色彩来渲染当前物体，并设置 Alpha（透明度）值为 0。

**Exclude Mist:** 使当前物体不参与迷雾效果的渲染。

**Invert Z Depth:** 反转当前物体在渲染时的 Z 深度值。

**Light Group:** 使物体仅被指定的灯光组照明，在下面的选项框中可输入已建立的灯光组名称。

**Face Textures/Alpha:** 使用贴图的色彩和透明度，来覆盖材质的漫反射和 Alpha 值设置。

**Vertex Color Paint/Light:** 同样地，使用顶点着色的数值，来替换材质的漫反射参数设置。

**Object Color:** 调整物体的显示色彩，常用于游戏模式。

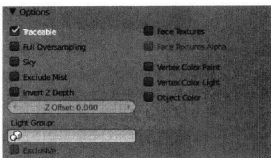


图 6-8 Options（选项）的参数面板

### 6.1.3 多材质应用

可以利用前面讲到的顶点组功能，为一个 Object 物体赋予多个材质。首先在物体模式中单击 Shift + A 添加一个球体，然后单击 Tab 进入其编辑模式。为物体新建两个顶点组，使用 Up 和 Down 来命名，并分别 Assign（标识）球体的上半部分和球体的下半部分，具体设置如图 6-9 所示。

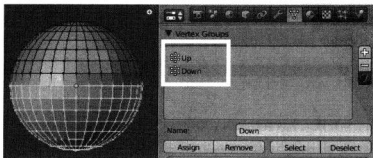


图 6-9 为物体新建两个顶点组

接着单击 Select 按钮选择 Down 顶点组，然后转入材质面板，单击 Add（添加）一个新的材质，设置颜色为红色。系统默认添加材质的方式是自动将其赋予整个物体，所以 Up 顶点组的网格点也显示为红色。单击 Assign 将材质赋予选中的 Down 顶点组，如图 6-10 所示。

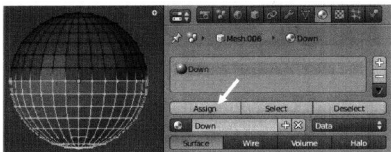


图 6-10 赋予 Down 顶点组材质

单击快捷键 A 取消选择，然后在顶点组面板中重新选择 Up 顶点组，再次回到材质面板，新建一个新材质 Up，将颜色修改为绿色。接着将其 Assign（赋予）Up 顶点组，如图 6-11 所示。可以看到，同一个物体上被应用了多个材质。

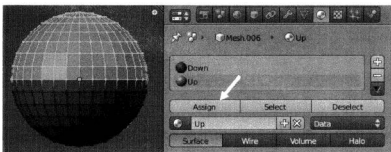


图 6-11 添加第二个材质

## 6.2 着色器选项

Blender 使用着色器的各项选项模拟真实物体的折射率、反射率和透明度等物理特性。

### 6.2.1 漫反射着色器

当光线照射到一个物体表面上时，表面上的每个点会把光线重新向四面八方反射出去，并且各反射光线的法线方向都不一致，这种反射称之为漫反射。漫反射意味着无论在哪个角度观察，视角上都会从表面上的该点接受到同样数量的光线，使得反射光的效果不受视觉角度的影响。但是表面上的反射光线数量将取决于入射角，假如入射光由于不同的入射角而产生了不均匀的散射，那么曲面将呈现不光滑的外表。如图 6-12 所示，为一个场景中灯光、物体和镜头的位置关系图。

Blender 提供了 4 种模式来计算漫反射。如图 6-13 所示，为漫反射着色器的参数面板。其中色彩面板用于选择漫反射的颜色，Intensity 用于控制反射的强度，Diffuse Shader Model 着色器类型用于选择参与运算的着色器类型。

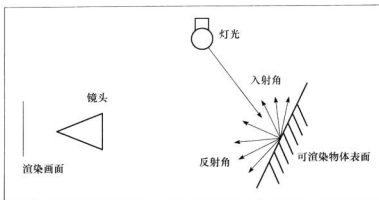


图 6-12 漫反射原理图

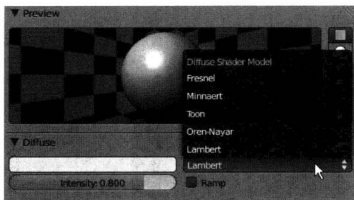


图 6-13 漫反射着色器的参数面板

### 6.2.1.1 Lambert (兰伯特)

这是 Blender 默认的漫反射着色器，也是最常用、最理想的漫反射类型。它能表现均匀散射的材质特点，对于不太粗糙的物体非常实用。Lambert (兰伯特) 不会反射周围的环境光线，但是当开启镜面属性后，材质表面将产生轻微的折射效果，所以常用于模拟木头和岩石等大自然中不平滑的材质物体。如图 6-14 所示，为 Lambert 的参数面板，其中 Intensity (强度) 可控制漫反射的力度，值越大，反射的强度也越大。

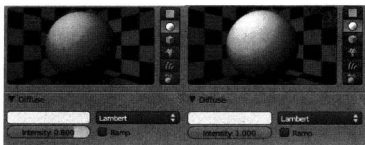


图 6-14 Lambert (兰伯特) 的参数面板

### 6.2.1.2 Oren-Nayar (奥伦-纳亚尔)

Oren-Nayar (奥伦-纳亚尔) 是一种更柔和的自然材质, 它看上去更平坦, 适合制作布料和陶器等物体。如图 6-15 所示, 为 Oren-Nayar 的参数面板, 其中 Roughness 用于设定表面的粗糙程度, 较大的值可以产生较轻微的漫反射效果。



图 6-15 Oren-Nayar (奥伦-纳亚尔) 的参数面板

### 6.2.1.3 Toon (汤恩)

Toon (汤恩) 是一种非常不自然的着色器, 它使用了清晰的光影边界和无衰减的光影区域, 产生特殊的类似卡通的表面效果, 所以常用它来制作卡通类的渲染。如图 6-16 所示, 为 Toon 的参数面板, 其中 Size 定义了光照区域的尺寸, Smooth 定义了阴影边界的平滑度。

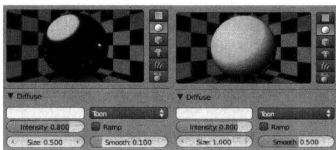


图 6-16 Toon (汤恩) 的参数面板

### 6.2.1.4 Minnaert (明纳尔特)

Minnaert (明纳尔特) 使用了和 Lambert (兰伯特) 相同的算法, 只是它添加了额外的参数来降低正常反射方向上的亮度。如图 6-17 所示, 为 Minnaert 的参数面板, 其中 Darkness (暗度值) 定义了物体的反向边缘亮度, 当值为 1 时, 材质将与 Lambert 的效果完全一样。较低的暗度值则会使物体整体变亮, 如图 6-17 左图所示; 较高的暗度值将会使得物体的视图中心变暗, 出现类似于绒毛的效果, 如图 6-17 右图所示。



图 6-17 Minnaert (明纳尔特) 的参数面板

### 6.2.1.5 Fresnel (菲涅尔)

Fresnel (菲涅尔) 是一种反射衰减现象, 当光到达材质接触面时, 一部分光被反射, 而另一部分将产生折射效果。所以直接接触光源的部分将显得较暗, 垂直于入射光的部分将变得较亮。如图 6-18 所示为 Fresnel 的参数面板, 其中 Fresnel 值用于控制菲涅尔效果的强度, Factor 值决定了混合渐变的大小。

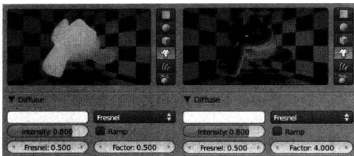


图 6-18 Fresnel (菲涅尔) 的参数面板

### 6.2.2 高光着色器

高光是在物体表面上亮度最高的部分, 通常用于表现物体的光滑质感。不同于漫反射, 高光的视觉效果是根据视图角度而定的, 如图 6-19 所示, 为一个场景中灯光、物体和镜头的位置关系图。这里要注意: 高光又不同于镜面反射光, 它仅显示物体表面区域的亮度效果, 而镜面反射是物体的本质现象, 从任何角度都能产生反射效果。

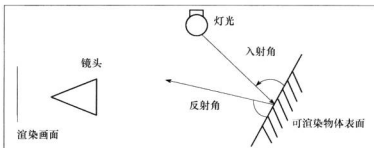


图 6-19 高光原理图

从本质上讲, 物体产生漫反射和高光效果的原理都是一样的, 它们根据的都是材质的物理特征和物体表面的微平面特征。只是漫反射是物体上表面不平滑的各点对光线向四面八方的反射效果, 而高光则是表面上的某一个区域上的点使光线向同一个方向上反射的效果。

Blender 提供了 5 种模式来模拟高光, 如图 6-20 所示, 为高光着色器的参数面板。其中色彩面板用于选择高光的颜色, Intensity 控制高光强度值, Specular Shader Model 着色器类型用于选择着色器类型。

#### 6.2.2.1 CookTorr (库克托)

CookTorr (库克托) 基于由 Robert L. Cook 和 Kenneth E. Torrance 于 1981 年提出的 Cook-Torrance (高光光照) 数学模型, 这是一种最常见的高光计算方式。CookTorr 是 Blender 默认的高光着色器, 常用于表

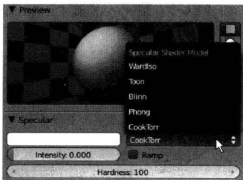


图 6-20 高光着色器的参数面板

现光亮的塑料表面。如图 6-21 所示，为 CookTorr 的参数面板，其中 Intensity 控制高光强度，Hardness（硬度）控制反射表面对高光的影响力度，值越大高光点越小。

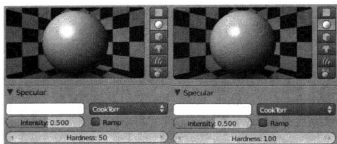


图 6-21 CookTorr（库克托）的参数面板

#### 6.2.2.2 Phong（冯）

Phong（冯）是基于由 Bui Tuong Phong 提出的利用插值来计算镜面反射光强的经验数学模型，该模型认为镜面反射的光强与反射和视线的夹角有关。Phong 的效果和 CookTorr 近似，但是高光点看上去更圆润，明暗界限更分明，适合于玻璃和水滴等光斑效果突出的材质。如图 6-22 所示，为 Phong 的参数面板，其中 Intensity 为光照强度，Hardness（硬度）控制反射表面对高光的影响力度，值越大高光点越小。

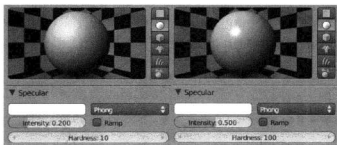


图 6-22 Phong（冯）的参数面板

#### 6.2.2.3 Blinn（布林）

James F. Blinn 在 Phong 模型的基础上提出了修正的数学模型 Blinn-Phong，它改进了 Phong 的高光，并混合了 Lambert 的漫反射部分和标准高光效果。它的渲染效果比 Phong 的高光更自然更柔和，因此适合一些具有清晰高光区域的物体，如金属等。由于省去了计算反射光线方向向量的两个乘法运算，所以它的运算速度也更快。如图 6-23 所示，为 Blinn 的参数面板，其中 Intensity 为光照强度，Hardness（硬度）控制反射表面对高光的影响力度，值越大高光点越小，IOR（折射率）则控制折射的力度。

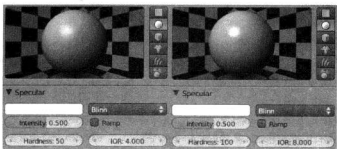


图 6-23 Blinn（布林）的参数面板



#### 6.2.2.4 Toon (汤恩)

Toon (汤恩) 的高光着色器和 Toon 漫反射着色器原理类似, 并可以相互搭配使用, 用于产生尖锐的卡通高光效果。如图 6-24 所示, 为 Toon 的参数面板, 其中 Size 定义了光照区域的尺寸, Smooth 定义了阴影边界的平滑度。

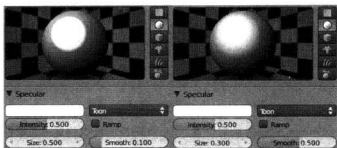


图 6-24 Toon (汤恩) 的参数面板

#### 6.2.2.5 WardIso (瓦多索)

WardIso (瓦多索) 是一个相对于其他着色器更灵活的高光着色器, 更适合用于金属和塑料物体。如图 6-25 所示, 为 WardIso 的参数面板, 其中 Intensity 为光照强度, Slope (斜率值) 越大高光过度越平滑。



图 6-25 WardIso (瓦多索) 的参数面板

### 6.2.3 渐变着色选项

Ramp (渐变着色) 选项允许控制材质上色彩的渐变范围, 以及和多种颜色混合的输出效果。Ramp 着色器不仅可用于漫反射着色器, 也可用于高光着色器。如图 6-26 所示, 为 Ramp (渐变着色) 选项的参数面板。

**Add/Delete:** 默认的色带中只有两个颜色控制杆, 单击 Add 按钮可添加一个新的控制杆, Delete 可删除选中的颜色控制杆, 旁边的数字框用于显示当前选中的颜色控制杆编号。

**Interpolation:** 插值方式, 这里可选择 5 种模式来计算多色彩之间的过渡效果, 包括 Ease (缓和)、Cardinal (鲜艳)、Linear (线性)、B-Spline (B 样条曲线) 和 Constant (常值) 模式。如图 6-27 所示, 从左至右分别为 Ease、B-Spline 和 Constant 的色带效果。

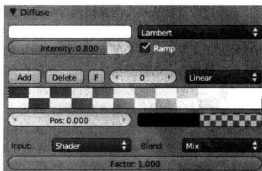


图 6-26 Ramp (渐变着色) 选项的参数面板



图 6-27 Interpolation (插值) 模式

**Position/Alpha:** Position 控制滑块的位置, Alpha 则控制该点位置上色彩的透明度。

**Input:** 这里用于选择色带在材质上的输入通道, 其中 Shader 选项将色带添加到着色器效果, Energy 将色带添加到灯光强度的效果, Normal 将色带添加到表面法向效果, Result 则将色带覆盖至着色器的输出效果。

**Blend:** 混合方式, 包括 Mix (混合)、Add (添加) 和 Multiply (叠加) 等。

**Factor:** 用于控制渐变着色器的输出在材质上的混合效果。

分别渲染图 6-27 中的渐变设置, 效果如图 6-28 所示。



图 6-28 渐变着色渲染效果

## 6.2.4 光线追踪反射

光线追踪就是从视角向视图做反向光线跟踪的过程, 那么当光线往回碰到无反射属性的物体表面时, 就会返回这点的颜色属性。如果这一点存在反射属性, 那么光线将在此点上向四周的环境方向上发散。如果接下来发散光线接触的第二个物体上无反射属性, 那么光线将返回当前第二个物体上点的颜色属性, 反之光线则会持续地反向追踪下去。理论上讲, 如果所有的物体都具有可反射的属性, 那么光线将在场景内做无法停止的持续反射计算。

但是现实世界中光线是有衰减的, 因为当光线接触到一个物体时, 一部分能量会被吸收, 只有一部分能被反射, 而人眼只能感觉到反射的能量。在 Blender 的默认渲染环境中, 光线是没有能量衰减计算过程的, 因此需要使用 Depth (追踪深度) 等参数来控制光线的反向追踪距离以及允许反弹的次数。如图 6-29 所示, 为光线追踪反射的参数面板。

**Reflectivity:** 反射强度, 数值越大镜面反射的强度也越大。一般来说反射的颜色都是白色, 对于一些特殊效果, 也可以单击调色板来修改反射的颜色。如图 6-30 所示, 左图为 Reflectivity 值是 0.5 和白色反射光的渲染效果, 右图为完美级反射 Reflectivity 值为 1 和橙色

反射光的渲染结果。

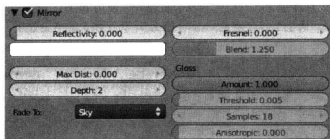


图 6-29 光线追踪反射的参数面板



图 6-30 Reflectivity (反射强度) 和不同颜色反射光的效果

**Depth:** 反射深度, 用于控制光线的反弹次数, 一般来说可设置为 2。

**Max Dist:** 反射计算的初始距离, 当物体到摄像机的距离小于这里设置的值时, 渲染器将不计算这个范围内物体上的反射效果。

**Fade To:** 对于不计算反射效果的物体, 可在这里设置它们的反射效果为过渡模式, 其中 Fade to Sky 是将其过渡到环境设置中的天空颜色, Fade to Material 则是过渡至材质本身的效果。如图 6-31 所示, 左图为 Max Dist 为 0 的渲染结果, 这时距离摄像机 0 个 Blender 单位以外的物体都将产生反射效果。右图为 Max Dist 为 0.1 的渲染效果, 物体与摄像机的距离刚好处于这个范围内, 由于选择的是 Fade to Sky, 那么物体表面的反射将采用材质到天空的过渡色彩。

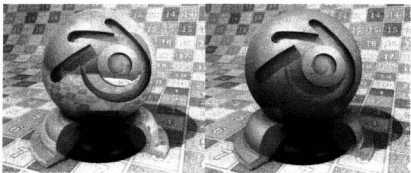


图 6-31 反射距离对反射效果的影响

**Fresnel:** 菲涅尔参数, 用于定义菲涅尔效果的影响系数。菲涅尔效果取决于表面法向与视角的角度, 一般来说, 角度越大, 材质上的反射量越大, 使用 Blend (混合) 选项来控制混合的强度。如图 6-32 所示, 从左至右分别是 Fresnel 值为 0、2.5 和 5 的渲染效果。



图 6-32 Fresnel (菲涅尔) 效果

**Gloss:** 光泽效果, 光泽度越高的物体表面越光滑, 反之低光泽度的物体表面看上去会更粗糙, 反射阴影更模糊。Gloss 效果能提供十分真实的自然表面, 但是如果 Gloss 设置过低, 渲染时系统将消耗更多的时间和资源, 所以请慎用 Gloss 参数。

**Amount:** 光泽强度, 当值为 1 时物体表面的平滑度最高。

**Threshold:** 阈值, 用于控制采样自适应性能, 值越低采样越频繁, 而且噪点越小。

**Samples:** 采样值, 值越高表面越光滑。

**Anisotropic:** 各向异性过滤, 用来过滤因为使用了较低的 Gloss 参数, 而在粗糙表面上产生的纹理反射错误。如图 6-33 所示, 从左至右分别是在 Gloss Amount 为 0.8 的基础上, 修改 Anisotropic 为 0 和 1 的渲染效果。

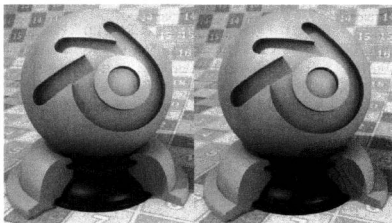


图 6-33 物体的 Gloss 不平滑反射效果

### 6.2.5 光线追踪透明

光线追踪也可用于完成物体的透明效果计算, 在光线从视角向视图做反向光线跟踪时, 当往回碰到有透明属性的物体表面时, 光线就会穿过物体继续向前进。当碰到一个无透明属性的物体时, 光线则会返回此点上的颜色属性, 光线追踪也将停止。产生材质的透明效果有 2 种模

式,如图 6-34 所示,为 Transparency (透明) 属性的参数面板。



图 6-34 Transparency (透明) 属性的参数面板

#### 6.2.5.1 Z Transparency (深度) 模式

Z Transparency (深度) 模式将直接使用 Z 深度值来控制物体的透视效果,它仅产生最简单的透明效果,没有折射率等高级控制参数。

**Alpha:** 材质的透明度。

**Specular:** 高光的亮度控制。

**Fresnel:** 菲涅尔效果控制参数。

**Blend:** 菲涅尔效果的混合控制参数值。

如图 6-35 所示,左图为没有使用 Fresnel 参数的透明效果,而右图为 Fresnel 参数为 1.2 时的 Z Transparency (透明) 渲染效果,与左图相比较其表面的混合效果更好。

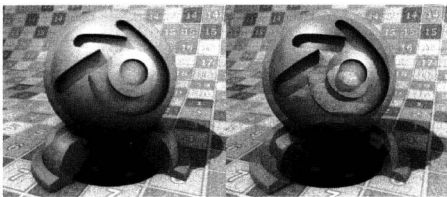


图 6-35 使用 Fresnel 的 Z Transparency (透明) 渲染效果

#### 6.2.5.2 Raytrace (追踪) 模式

如图 6-36 所示,为 Raytrace (光线追踪) 的透明模式参数面板。Raytrace 在 Z Transparency 单纯的透明度控制基础上,添加了更多细节设置,并使用光线追踪的算法来模拟透明物体的物理属性。

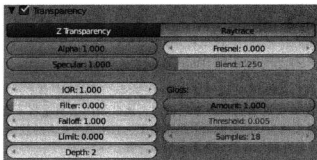


图 6-36 Raytrace (光线追踪) 的透明模式参数面板

**IOR:** 物体的折射率,不同物体的折射率属性也不同。如表 6-1 所示,为一些常用物体的折射率。

表 6-1 常用物体折射率列表

材质	IOR 折射率	材质	IOR 折射率
银	0.180	紫水晶	1.544 ~ 1.553
黄金	0.470	盐	1.544
空气	1.000	石英	1.550
气泡	1.100	糖	1.560
冰	1.310	无色玻璃	1.613
酒精	1.329	黄玉	1.620 ~ 1.627
水	1.333	沥青	1.635
塑料	1.400	打火石	1.650
铝	1.440	翡翠	1.660 ~ 1.680
玻璃	1.450	花岗岩	1.760
黑曜石	1.480 ~ 1.510	红宝石	1.760 ~ 2.419
黑玛瑙	1.486 ~ 1.658	水晶	2.000
丙烯酸玻璃	1.491	铅	2.010
甘油	1.473	钻石	2.417 ~ 2.541
汽油	1.501	铬	2.420
铁	1.510	钢	2.500
尼龙/贝壳	1.530	铜	2.705
碧玉/象牙	1.540	赤铁	2.940

**Filter:** 过滤值, 用于调节追踪计算时的光线过滤数量, 值越大光线的穿透率越低。如图 6-37 所示, 左图为 Filter 为 0 时的效果, 右图为增大 Filter 到 1 的渲染效果, 可以看到物体的透明度明显下降了。

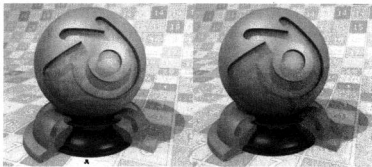


图 6-37 Filter 调节光线的过滤数量

**Falloff:** 衰减量控制, 值越高光线的衰减越快, 这能为玻璃等物体产生厚度和深度的效果。

**Limit:** 当物体厚度大于这里的设置极限值时, 该区域内的材质将不会产生透视效果。如图 6-38 所示, 左图为极限值为 0 时的渲染效果, 右图为极限值为 0.5 的效果, 可以看到厚度大于这个值范围的物体表面部分都没有参与透明效果的计算。

**Depth:** 光线的穿透深度, 当光线到达设置的深度值时将不再向前追踪, 后面的物体将显示为黑色或材质本色。所以当发现透视的物体表面部位在渲染后呈现黑色时, 可以适当增加 Depth



图 6-38 Limit 调节物体的透明度效果

(深度) 值来完善物体的透视追踪效果。如图 6-39 所示, 左图为 Depth 为 0 的效果, 可以看到光线无法穿透物体上厚度较大的部分, 而右图为 Depth 为 4 的效果, 可以看到大部分光线都可以穿透整个物体, 并产生了透明效果。



图 6-39 Depth 定义光线的追踪穿透深度

**Gloss:** 真实的透明物体 (例如玻璃) 并不是一个完美的光滑介质, 光线会有一部分能量被表面的反射吸收掉。这里可以通过调节 Gloss (光泽度) 来控制透明物体的表面属性, 用于模拟真实介质上的非完全穿透效果。如图 6-40 左图所示, 为正常完美的表面渲染效果, 将 Amount (光滑度) 降为 0.8, 如图 6-40 右图所示, 物体表面出现了明显的磨砂不平滑效果, 并减弱了光线的穿透追踪深度。

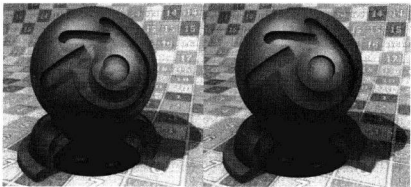


图 6-40 Gloss 控制物体的表面光滑度

### 6.2.6 子面散射着色器

Subsurface Scattering (子面散射着色器, SSS) 是一种能够模拟物体内部散射光线作用的特殊材质, 在透明或半透明的材质上, 这种效果尤为明显, 例如果冻、葡萄和皮肤等。SSS 是一种很难模拟的物理效果, 因此它也是高质量真实渲染中最复杂的材质之一。它的物理真实度需要经过大量的计算机演算, 才能使散射比例达到一个柔和的效果。如图 6-41 所示, 为艺术家 Lucas Veber 制作的 SSS 渲染效果。

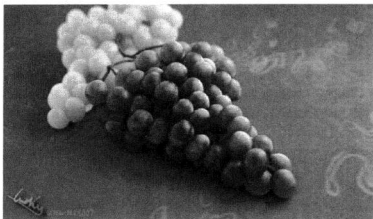


图 6-41 Lucas Veber—Grapes

事实上由于物体本身具有的复杂各向异性和不均匀密度等特性, 直接计算物体表皮下的光线路径是不可能的事情, 如果真要运算物体内部光子的反射, 那将是一个十分复杂的微分数学方程, 求解也需要花费大量的计算时间。为了得到较快的速度和近似的效果, 可以使用其他一些算法来模拟这种效果。

在 Blender 中, 当物体接受光照时, 系统首先会分别计算出物体前后方的光照强度和物体表面上的亮度。这如同法向式渲染——利用光线的深度值来近似模拟光线可穿透的物体厚度, 这样就使得在厚度越薄的位置, 光线穿透力将越强。接下来系统将使用 SSS 着色器来替代漫反射着色器, 计算光线对物体表面的反射影响效果, 通过综合物体表面的亮度值和光线深度值的平均值, 来确定最终的表面光照强度值, 以得到较真实的散射效果。如图 6-42 所示, 为 SSS 的参数面板, 这里提供了多种默认材质的着色器, 单击 Presets 可在下拉菜单中做快速选择。

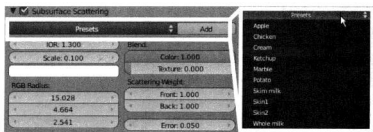


图 6-42 SSS 的参数面板

**IOR:** 折射率控制着入射光的衰减程度, 在较高的折射率下, 光线的衰减也越快。IOR 对柔和效果的变化是很微妙的, 一般来说设置为 1.3 ~ 1.5 比较合适。如果需要模拟特定的物体, 可以参考表 6-1 中的 IOR 列表。



**Scale:** 物体的缩放比例, 由于需要使用光线深度值来模拟物体的厚度, 所以设置物体大小与环境的比例很重要。例如设置 Scale 为 1 时, 1 个 Blender 单位的距离就相当于 1 毫米的实际长度, 而当设置 Scale 为 0.001 时, 1 个 Blender 单位距离就约等于 1 米。

**RGB Radius:** 这里分别控制 RGB (红、绿、蓝) 三色光线在物体中的散射半径。由于光线穿越物体时会产生一定的光线路径, 那么控制这些色彩的散射半径可改变物体散射表面的颜色效果。单色通道中的值越大, 产生对应的颜色影响效果越强。

**Blend:** 混合参数值, 用于修改物体本身着色器的颜色与 SSS 的混合效果。

**Scattering Weight:** 散射权重, 实际上就是分别控制光线的前后穿透强度, 当值为 1.0 时, 代表穿透强度为 100%。

**Error:** 容差, 调整容差可控制算法的采样率, 以及物体表面的仿真精确度, 默认值 0.5 可提供较好的无瑕疵效果。当增大 Error 时, 渲染时间能减少, 但是可能会产生一些错误的计算效果。不过在为材质做测试预览时, 可以将 Error 设置为 1, 以快速预览修改结果。

接下来通过模拟葡萄的 SSS 效果范例, 来讲解各种参数的使用方法。首先修改漫反射和高光着色器, 将漫反射颜色调为紫色, 着色器设置为默认的 Minnaert。接着添加 Ramp 效果, 如图 6-43 左图所示。然后设置高光色为白色, 使用 Phong 着色器, 添加一个色彩过渡, 如图 6-43 右图所示。

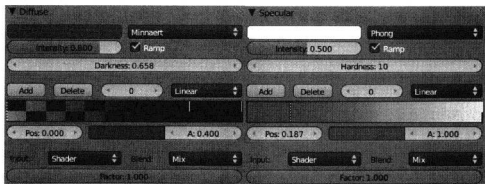


图 6-43 漫反射和高光着色器设置

接下来开启 SSS 参数计算, 修改颜色和 Blend 混合值。在 RGB Radius 中适当地增加红色散射值, 调大 Back 背面的透光性, 具体参数如图 6-44 所示。

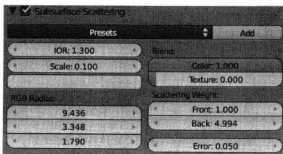


图 6-44 SSS 参数设置

最后单击渲染, 如图 6-45 右图所示, 为了对比 SSS 对物体的影响, 左图为取消 SSS 的渲染结果。可以发现, 添加了 SSS 效果的葡萄表皮产生了轻微的透明效果, 由于强化了红色光线的穿

透力，所以物体表面渗透出淡红的散射效果。对比没有 SSS 效果的材质，右图的渲染结果看上去更生动更真实了。



图 6-45 SSS 渲染效果的对比

### 6.2.7 体积着色器

在制作烟雾和云等特殊物体时，需要设置材质为 Volume（体积）着色模式。单击材质面板，设置模式为 Volume 即可。如图 6-46 所示，为 Volume（体积）着色器的预览效果。

在 Surface（表面）着色时，系统需要计算光线的跟踪路径，而体积着色则是计算光线在穿过物体时，与物体中抽象的“颗粒”产生反弹和衰减吸收的效果。同为透明的效果，体积着色又不同于透明光线的追踪计算：透明物体的光线追踪，强调的是一种由传播方式产生的效果，例如折射和反射等光线的交互；而体积着色则强调的是物体内部的光线属性变化，包括像素体积化后的色彩变化等。如图 6-47 所示，为 Volume（体积）着色器的参数面板。

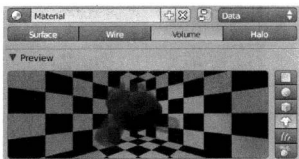


图 6-46 Volume（体积）着色器的预览效果

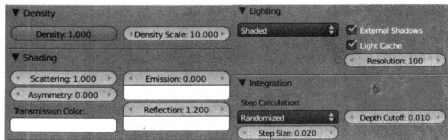


图 6-47 Volume（体积）着色器的参数面板

**Density:** 密度值，当设置较大的密度值时，物体的浓度越高，光线越难以穿过。如图 6-48 左图所示，为 Density 值为 2.5 的渲染效果，而右图为 Density 值为 10 的渲染效果。

**Scattering:** 散射值，用于控制光线在物体中的散射效果。低散射值将使较多的光线随意散射，物体将变得更透明，同时物体的阴影将显得更轻。配合 Asymmetry（不对称）参数，可调整

前后方向上的光线散射效果。如图 6-49 左图所示，为设置 Scattering 为 0.5 的效果，而右图是设置 Scattering 为 5 的效果。

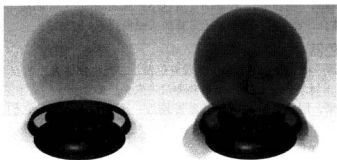


图 6-48 Density（密度值）渲染效果

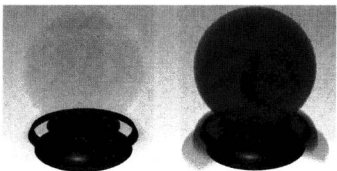


图 6-49 Scattering（散射值）对透光性的影响效果

**Transmission Color:** 传递光线的色彩，用于设置穿透物体后的光线色彩。如图 6-50 左图所示，为传递色彩为白色的默认渲染效果，而右图为修改色彩为橙色的渲染效果。

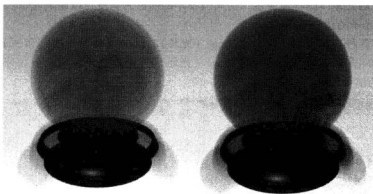


图 6-50 Transmission Color（光线传递）后的色彩效果

**Reflection:** 反射光线设置，用于配合 Scattering（散射值）一起调整光线在反射角度上的颜色，以及阴影的颜色效果。

**Emission:** 发光效果，例如在制作火焰时，开启发光功能可以使光线在穿透效果的基础上，产生一些自发光效果。

**Shading:** 着色模式, 可以选择使用最为简单、运算最快的 Shadeless (无着色) 模式, 也可以使用效果最好、但最花费时间的 Shaded + Multiple Scattering 组合计算模式。介于两者性能之间的模式还包括 Shadowed (阴影型)、Shaded (着色型) 和 Multiple Scattering (多重散射型) 等。

**Integration:** 整合计算功能, 用于修改计算的采样模式等。

## 6.2.8 光晕着色器

Halo (光晕) 是一种非面着色的特殊材质着色器, 它能将物体表面的点渲染出光晕的效果。如图 6-51 所示, 单击材质面板上的 Halo 选项即可进入光晕参数面板。从预览图上可以大致看到物体表面的光晕效果, 网格上的每个点都被转换为一个发光点, 但是它们并没有照明物体的功能。

### 6.2.8.1 Halo (光晕) 参数面板

光晕的参数面板和普通材质参数面板相似, 它新增了两个特殊参数 Halo (光晕) 和 Flare (耀斑) 面板。如图 6-52 所示, 为 Halo (光晕) 的参数面板。

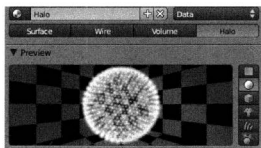


图 6-51 Halo (光晕) 效果预览

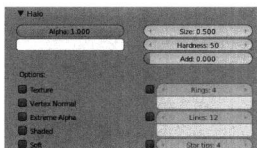


图 6-52 Halo (光晕) 的参数面板

**Size/Hardness/Add:** 分别控制光晕的尺寸大小、硬度和强度。

**Texture:** 是否赋予光晕材质。

**Vertex Normal:** 使用点的法线来控制光晕的尺寸大小。

**Extreme Alpha:** 在光晕的末端使用 Alpha (透明) 效果。

**Shaded:** 开启光晕体的环境着色, 使周围物体的阴影和亮度可以影响光晕的效果。

**Soft:** 柔化光晕的边缘效果。

Halo 总共有 3 种光晕模式, 分别为 Rings (光环)、Lines (放射光线) 和 Star (星光)。

☐ **Rings:** 如图 6-53 所示, 为开启 Rings (光环) 的光晕效果, 从左至右分别为 2 环、4 环和 6 环的渲染效果。

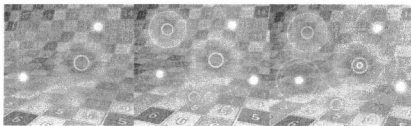


图 6-53 Rings (光环) 的渲染效果

☐ **Lines:** 如图 6-54 所示, 为添加 Lines (放射光线) 的效果, 从左至右分别为 3 剑、6 剑和 16 剑的渲染效果。



图 6-54 Lines（放射光线）的渲染效果

- **Star**：为光晕的 Star（星形）效果，如图 6-55 所示，从左至右分别为 3 星、8 星和 16 星的渲染效果。

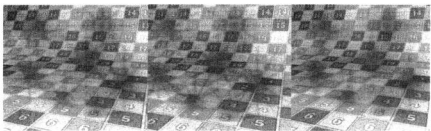


图 6-55 Star（星形）的渲染效果

#### 6.2.8.2 Flare（耀斑）参数面板

如图 6-56 所示，为 Flare（耀斑）的参数面板。

**Size**：用于控制 Halo 的耀斑尺寸，值越大耀斑越大。

**Boost**：为耀斑添加一些额外的长度效果。

**Seed**：为光晕和耀斑之间添加一些随机偏移量。

**Subflares**：设置子耀斑的数量。

**Subsize**：设置子耀斑的尺寸。

**Shading**：设置材质的透明属性。

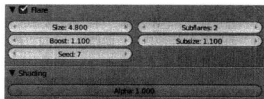


图 6-56 Flare（耀斑）的参数面板

### 6.2.9 顶点着色器

Blender 提供了一套 Vertex Color（顶点着色）系统，这是一种与材质无关的色彩数据信息，它可以直接被硬件计算并显示出来，常被应用在游戏等实时渲染中，效率很高。顶点着色的效果和网格的面数量有关，当网格上的点越多时，可着色的顶点数也就越多，产生的效果越好。

在选择一个物体后，单击状态模式，切换至 Vertex Paint（顶点绘制）模式。如图 6-57 所示，为 Vertex Paint（顶点绘制）模式的窗口面板。

顶点绘制模式和 Texture Paint（贴图绘制）模式十分类似，单击 T 弹出绘制菜单，在调色板中选取颜色后即可直接在物体上进行绘制了。对同一个物体，可以绘制多个顶点色，在物体属性中，Vertex Colors 提供了多顶点着色层的管理，旁边的加减符号用于新建和删除。如图 6-58 所示，为 Vertex Colors（顶点色）管理面板。

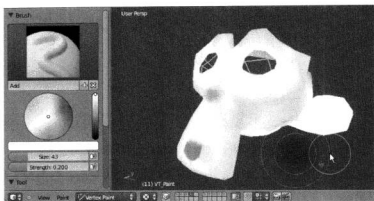


图 6-57 Vertex Paint（顶点绘制）模式的窗口面板

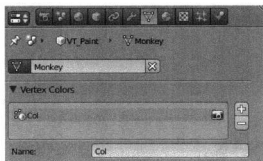


图 6-58 Vertex Colors（顶点色）的管理参数面板

经过顶点着色的物体，可以不需要添加任何材质和纹理，即可直接在 3D 视图中显示色彩信息，如图 6-59 左图所示。同样地，也可以添加一个材质，并开启 Option（选项）中的 Vertex Color 选项，使用 F12 直接渲染，效果如图 6-59 右图所示。

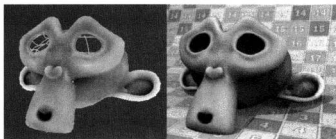


图 6-59 顶点着色物体的两种显示效果

### 6.3 纹理贴图

物体表面的质感由着色器来模拟，而纹理和花纹就可以由 Texture（贴图）来完成。贴图不仅能为物体添加更多的色彩和阴影等细节，还能通过特殊的映射方式带给物体表面更多的造型效果。贴图可以被理解为一层层包裹在物体表面上的图层，就像洋葱皮一样，每一个图层都可以产生不同的影响效果，用于改变物体表面对应的属性，例如色彩、高光和凹凸感等。

### 6.3.1 通用贴图参数

贴图属性是建立在材质的基础上，而不能建立在物体本身。当为物体添加了一个材质后，即可在材质面板的旁边找到贴图面板，如图 6-60 左图所示。单击 Add（添加）可为物体新建一个贴图，接着面板会弹出 Preview（预览）窗口，和着色器面板类似，可以选择多种预览方式，如图 6-60 右图所示。



图 6-60 Texture（贴图）的参数面板

可以在贴图列表表中为当前材质添加多个贴图图层，图层的管理和 Photoshop 等软件类似，旁边的小图标能显示一个快速的预览效果，如图 6-61 所示。每个贴图图层都有单独的映射通道，单击图层旁边的选项框即可控制是否启用该贴图通道层。这里要注意：图层的顺序对最后的效果影响很大，物体在渲染时是从上向下计算贴图层的，如果两个图层都指向颜色通道，那么第二个图层 Text02 就会覆盖第一个图层 Text01 的效果。可以使用面板旁边的箭头来更改纹理图层的顺序。



图 6-61 图层管理

Blender 提供了 15 种贴图类型，如图 6-62 所示，下面简单介绍其中最常用的几种类型。



图 6-62 Type（贴图类型）菜单

#### 6.3.1.1 Blend（混合）类型

Blend（混合类）是最常用的贴图之一，常用于毛发或多重纹理之间的渐变效果。如图 6-63 所示，为 Blend 材质的参数面板。

这里提供了多种混合模式，包括 Linear（线性化型）、Quadratic（二次型）、Easing（宽松型）、Diagonal（对角线型）、Spherical（球型）和 Radial（放射型）。如图 6-64 所示，从左至右分别为这些

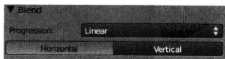


图 6-63 Blend（混合）类型的参数面板

混合模式的预览图效果。

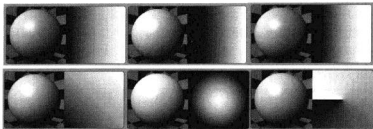


图 6-64 Blend (混合) 模式预览效果

### 6.3.1.2 Noise (噪点) 类型

Noise 可以产生随机的噪点图案,如图 6-65 所示,为默认的 Noise (噪点) 效果图。

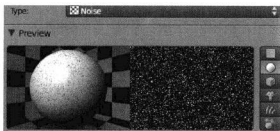


图 6-65 Noise (噪点) 类型

### 6.3.1.3 Wood (木纹) 类型

Wood (木纹) 贴图主要应用于木纹等条形纹理的模拟,如图 6-66 所示,为 Wood (木纹) 的贴图效果和对应的参数面板。



图 6-66 Wood (木纹) 的参数面板

**Sine/Saw/Tri:** 控制纹理的波长分别为正弦、锯齿和三角波的类型。

**Bands/Rings/Band Noise/Ring Noise:** 选择纹理的混合模式。

**Soft/Hard:** 用于控制噪波的硬度。

**Basis:** 用于调整噪波的大小,它可以产生一些特殊的效果。如图 6-67 所示,可以利用 Basis 值制作出一些有趣的效果。

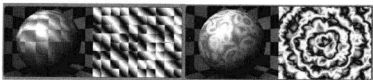


图 6-67 特殊噪波对纹理的影响



#### 6.3.1.4 Clouds (云) 类型

Clouds (云) 纹理, 常用于产生火焰和烟雾的效果, 如图 6-68 所示, 为 Clouds (云) 纹理的参数面板。



图 6-68 Clouds (云) 的参数面板

**Greyscale/Color:** 控制云的花纹为单色还是多色。

**Noise:** 控制云边缘的柔和程度。

**Basis:** 控制云的 Size (大小) 和 Depth (深度), 适当调整这些参数, 也能制作出许多有趣的纹理, 如图 6-69 所示。

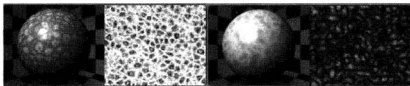


图 6-69 特殊的 Clouds (云) 纹理效果

#### 6.3.1.5 Image (图片) 类型

使用图片作为贴图的纹理, 是一种最快速简单的纹理输入方式。如图 6-70 所示为 Image 的参数面板。



图 6-70 Image (图片) 的参数面板

Blender 支持几乎所有的图片格式贴图, 单击 Image 面板中的文件夹按钮, 即可打开图片浏览器, 选择图片文件后, 系统面板中会自动弹出 Mapping (映射) 和 Sampling (采样) 面板, 如图 6-71 所示。

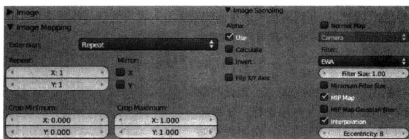


图 6-71 Mapping (映射) 和 Sampling (采样) 参数面板

**Extension:** 扩展模式, 默认贴图排列为 Repeat (重复) 模式, 单击下拉菜单还可以选择 Extend (拉伸) 等模式。

**Repeat:** 是否复制纹理, 其中 X 和 Y 分别控制在 X 和 Y 方向上的纹理复制数量。

**Mirror:** 选择是否开启镜像纹理效果。

**Crop Minimum/Maximum:** 控制纹理中心在坐标上的偏移量。

**Alpha:** 是否使用图片的 Alpha 通道数据。

**Normal Map:** 是否使用图片的色彩信息来制作表面的法向效果。

**Filter:** 设置用于对贴图做采样计算的过滤器类型, 一般保持默认即可。

### 6.3.1.6 Marble (大理石花纹) 类型

Marble (大理石花纹), 常用于制作岩石等自然效果, 如图 6-72 所示, 为 Marble 的参数面板。

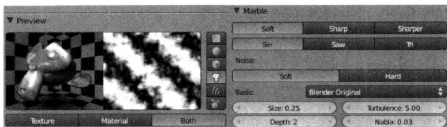


图 6-72 Marble (大理石花纹) 的参数面板

**Soft/Sharp/Sharper:** 设置纹理硬度, 纹理硬度越高, 边缘越清晰。

**Noise:** 控制噪波的硬度。

**Basis:** 调节噪波特性。

### 6.3.2 映射输入

一张贴图在进入贴图系统后, 首先需要被指定一个映射方式和一个映射坐标系, 这样才能被正确地输入至材质的着色流水线中。接着要再选择一个属性通道, 设置贴图将要影响的材质属性, 用于控制纹理的输出效果。如图 6-73 所示, 为一个贴图从输入到选择映射坐标, 最后输出至特定通道的工作流程图。

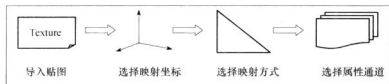


图 6-73 添加贴图的工作流程

其中, Mapping (映射) 方式就是根据不同的坐标系来设定纹理的投影方向, 这些坐标系可以来自各种空间。如图 6-74 所示, 为贴图 Mapping (映射) 输入方式的参数面板。

#### 6.3.2.1 Coordinates (映射坐标系)

**Global:** 让贴图使用全局的坐标系, 因此当移动物体时, 由于贴图已锁定为全局坐标, 所以不会随物体一同移动。如图 6-75 所示, 左图为初始位置的渲染结果, 右图为移动物体一段距离后的

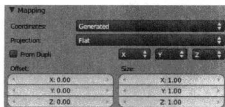


图 6-74 Mapping (映射) 输入方式的参数面板

渲染效果，可以看到物体的位移不会改变贴图的空间位置。

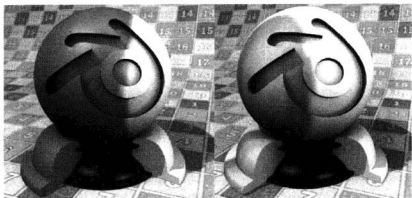


图 6-75 Global (全局) 坐标的映射效果

**Object:** 使用一个物体的坐标数据来定位贴图的坐标，一般来说常使用 Empty (空) 物体来定位贴图的坐标位置。如图 6-76 左图所示，首先在物体左前方添加一个 Empty 物体，然后在 Object (物体) 选项中输入 Empty 的对象名。当单击渲染后，可以看到 Logo 的中心将出现在球体的左上方。接下来将 Empty 物体移动到球体的右上角，如图 6-76 中图所示。再次单击渲染后发现，Logo 的中心随着 Empty 的移动发生了转移，效果如图 6-76 右图所示。

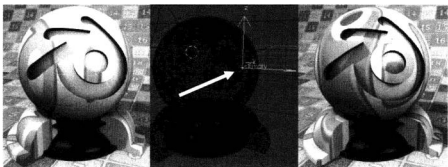


图 6-76 Object (物体) 坐标的映射效果

**Generated:** 使用物体的本地坐标，这也是物体映射的默认选项。

**UV:** UV 是将 2D 纹理投影在 3D 物体表面的一种精确映射方式。在使用 UV 贴图时，需要指定贴图映射采用本地的 UV 坐标系。如何制作本地的 UV 坐标系，将在后面的 UV 相关设置中详细讨论。

**Strand:** 将纹理映射到毛发物体上，需要结合 Strand 着色器选项来使用。

**Sticky:** 使用网格的粘度坐标系，来产生 UV 的贴图效果，这时贴图会根据镜头的移动而自动调整 UV，因此常用于制作 Camera Mapping (镜头映射) 等特殊贴图应用。

**Window:** 窗体坐标，这时贴图的映射方式将根据窗口尺寸而定。如图 6-77 左图所示，贴图为一个图案居中的正方形，那么渲染器将会直接把贴图映射到窗口的水平视角坐标上，结果如图 6-77 右图所示。

**Normal:** 法向坐标，将物体表面的法线矢量作为映射坐标系，常用于制作特殊视觉角度的场景。如图 6-78 左图所示，由于物体表面的法线分布不均匀，在法线错乱的位置处，贴图也将产生相应的拉伸，效果如图 6-78 右图所示。

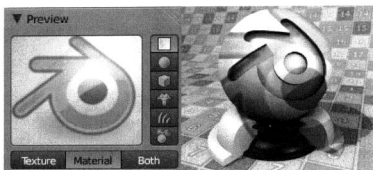


图 6-77 Window (窗体) 坐标的渲染效果

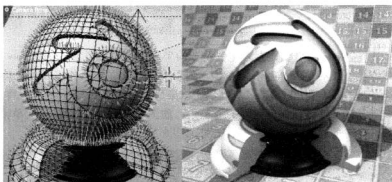


图 6-78 Normal (法向) 坐标的渲染效果

**Reflection:** 使用反射向量为贴图的坐标系，常用于为环境添加反射贴图。

**Stress:** 压力坐标，当网格物体受力变形产生挤压效果时，压力坐标可辅助制作表面材质的混合效果。例如橡胶被拉伸时，被拉伸的区域将显得更薄更透明。为了制作这种效果，可以使用一个线性的纹理，根据物体从压力坐标上读取出来的数据，将压力值转换为色彩值，并应用于透明遮罩通道中，制作出材质的透明过渡。如图 6-79 所示，为材质的结点设置。

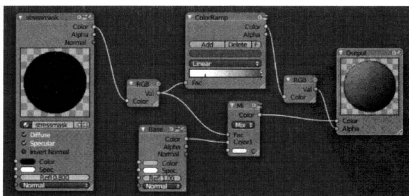


图 6-79 压缩材质的结点设置

在其中的 Streemask 材质里，添加一个 Blend (混合) 纹理，设置其模式为 Linear (线性化)，将映射坐标设置为 Stress，这样物体的压力数据就能以颜色数据的方式输出至下一个结点中，如

图 6-80 左图所示。最后在球体上添加一个 Plane 物体，设置其为 Soft（软体）属性，运行仿真模拟，当它向下运动被球体挤压出拉伸效果时，单击渲染就可以在材质上得到对应的弹性材质效果了，如图 6-80 右图所示。

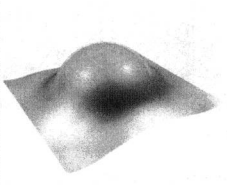
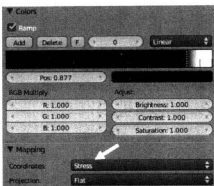


图 6-80 Stress（压力）坐标的材质应用

### 6.3.2.2 Projection（映射）方式

当完成了坐标系的选择，就要为当前的贴图指定一种 Projection（映射）模式，使 2D 图片能基于当前的坐标系，按照一定的映射方式将纹理画面投影至 3D 物体的表面上。贴图的映射模式一共分为 4 种。

- **Flat**：平铺，纹理将按照垂直于物体切线方向的坐标平面映射到表面上，对于 Sphere（球体）和 Plane（平面）物体的贴图效果较好，但是对于有直角面的物体，如 Cube 物体，它的切线方向会出现拉伸效果。如图 6-81 所示，从左至右分别为不同物体的 Flat（平铺）贴图效果。



图 6-81 Flat（平铺）映射的渲染效果

- **Cube**：立方体，如果说 Flat 最适合没有垂直边的物体，那么 Cube（立方体）映射最适合有大转角面的物体，如 Cube 和 Plane。如图 6-82 所示，从左至右分别为不同物体的 Cube（立方体）贴图效果。



图 6-82 Cube（立方体）映射的渲染效果

- **Tube**：顾名思义，Tube 将贴图按照管子的形状包裹在物体表面上，当制作酒瓶等物体

时,这种映射方式最适用。如图 6-83 所示,从左至右分别为不同物体的 Tube (管体) 贴图效果。

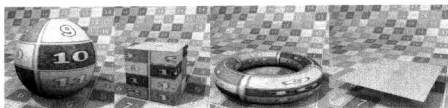


图 6-83 Tube (管体) 映射的渲染效果

- ❑ **Sphere:** 球面,最适合应用在球形物体上,例如行星等。如图 6-84 所示,从左至右分别为不同物体的 Sphere (球面) 贴图效果。



图 6-84 Sphere (球面) 映射的渲染效果

由于贴图的尺寸和物体存在偏差,所以纹理在表面经常出现重复和叠加。可以使用 Offset (偏移量) 来控制贴图的映射位置,其中 Offset X/Y/Z 可分别调节贴图在坐标轴 X/Y/Z 轴上的映射偏移量。同理,Size 也控制纹理的尺寸缩放变化。

### 6.3.3 通道输出

坐标选择和映射模式组成了贴图的输入 workflow,接下来就先了解一下系统对物体表面纹理的处理和属性影响的输出 workflow。

贴图不仅能控制物体的颜色等外观,还能将纹理应用到一些特殊的属性通道中,制作出类似 Bump (凹凸) 和 Normal (法向) 的效果。如图 6-85 所示,为输出纹理的通道属性面板。

这里每一个选项都代表一个独立的属性通道,通道的值用于控制属性在材质上的叠加效果。单个纹理可以被同时作用于多个通道属性中,但是建议在使用时,应分别针对不同的属性建立对应的纹理层。使用同一个纹理去影响多个属性通道,将会降低属性之间的相互作用性,而且这样也不利于纹理的管理和调节。

#### 6.3.3.1 Diffuse (漫反射) 通道

**Intensity:** 控制贴图的漫反射强度。

**Color:** 纹理的颜色对物体表面的色彩影响强度。如图 6-86 左图所示,是 Color 值设置为 1 的渲染结果,将 Color 值减小到 0.5,可以看到纹理在物体表面的颜色变浅了,如图 6-86 右图所示。

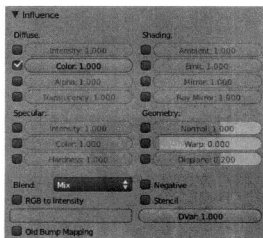


图 6-85 纹理的通道属性面板

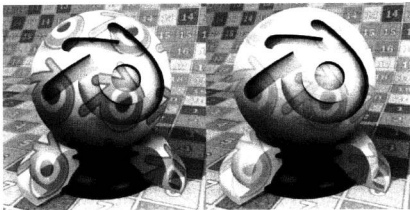


图 6-86 Color 通道值变化对渲染结果的影响

**Alpha:** 使用纹理的 Alpha 值来控制物体的透明属性。

**Translucency:** 设置纹理的透明度。

#### 6.3.3.2 Specular (高光) 通道

**Intensity:** 用于设置纹理对物体高光的影响强度。

**Color:** 使用物体纹理来控制高光区域的颜色，常使用 HDR 贴图来制作物体的高光。如图 6-87 左图所示，是未使用高光通道的渲染效果，当添加一个 HDR 贴图，使用其高光纹理来影响纹理的输出，效果如图 6-87 右图所示。使用高光通道的好处是，可以制作出与环境无关的高光效果，常应用于游戏等只需要在某一个物体上产生特殊高光的场景。

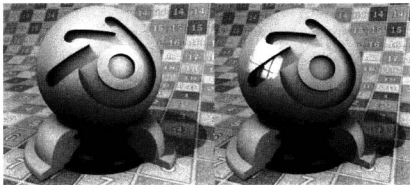


图 6-87 Specular (高光) 通道的纹理效果

**Hardness:** 控制高光的硬度通道。

#### 6.3.3.3 Shading (着色器) 通道

**Ambient:** 使用纹理来调节着色器上环境光的影响属性。

**Emit:** 使用纹理来调节着色器的自发光属性。

**Mirror:** 使用纹理来调节着色器的镜面反射强度。如图 6-88 左图所示，为使用默认的天光和環境光制作物体表面的反射效果，如果为当前材质添加一个 HDR 纹理，并将其映射至 Mirror 通道，效果如图 6-88 右图所示。可以看到如果在 Mirror 通道中应用贴图，可以叠加环境本身对物体的反射效果，制作出特殊的表面反射效果。

**Ray Mirror:** 将纹理添加到着色器的光线追踪计算中。

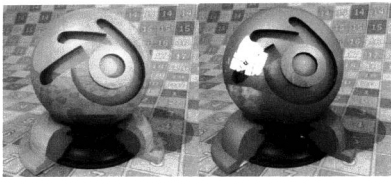


图 6-88 Mirror (镜面) 反射通道的纹理效果

#### 6.3.3.4 Geometry (多边形) 通道

**Normal:** 产生法向的贴图效果, 常将 Bump 贴图和 Normal 贴图指向这个通道, 用于产生多边形的表面凹凸效果。

**Warp:** 弯曲变形, 使用当前的贴图去影响下一个图层的贴图坐标数据。

**Displace:** 置换通道, 用于修改置换贴图的属性。置换贴图的原理请参考 4.2.4 中介绍的置换修改器。通道和修改器的应用区别仅在于, 通道只会在渲染结果中显示置换效果, 而修改器则可以实时地预览置换效果, 并将效果应用在最后的渲染中。对比 Normal 通道的贴图效果, 可以发现 Displace 置换效果更突出对网格的整体置换属性, 而 Normal 贴图着重于细节显示。如图 6-89 左图所示, 为将一个 Marble 默认纹理映射至 Normal 通道值为 1.5 时的效果, 而右图是将同样的纹理映射至 Displace 通道值为 0.1 时的渲染效果。



图 6-89 Normal 与 Displace 通道的对比渲染效果

#### 6.3.3.5 General (通用) 属性

**Blend Mode:** 混合模式, 用于选择纹理图层之间的叠加效果模式。

**RGB to Intensity:** 将纹理的色彩数据转换为灰度值, 用来计算叠加的强度效果。

**Negative:** 将纹理的色彩反转。

**Stencil:** 定义该图层为蒙版图层, 这和 Photoshop 中的蒙版原理一样。如图 6-90 所示, 将一个黑白两色的蒙版图叠加在原贴图图层上, 并开启其 Stencil 功能, 最后即可得到一个混合后的遮罩效果。

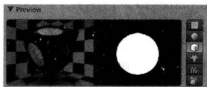


图 6-90 Stencil (蒙版) 图层



**DVar**: Destination Value (叠加值), 是通道输出的整体叠加效果控制参数。例如着色器的 Emit (自发光) 属性为 0, 那么当把贴图输出到 Emit 通道时, 如果此时设置该图层的通道 DVar (叠加值) 为 1, 那么最终渲染时系统将采用这两个参数之间的加权值 1 来计算物体的 Emit (自发光) 属性。如果设置 DVar 为 0, 那么纹理将不会对通道属性产生任何的叠加效果。

## 6.4 UV 贴图

UV 贴图是将 2D 的图片或纹理, 按照自定义的 UV 坐标系, 映射到 3D 物体表面的一种投影方法, 而 UV 拆分就是将 3D 物体按照 2D 坐标分离出 UV 坐标系的过程。例如常用的世界地图, 就是将地球表面拆分成了平面图形, 这个地图就是地球物体的 UV 贴图, 如图 6-91 所示。

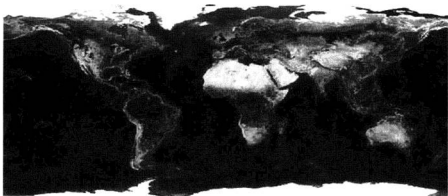


图 6-91 地图就是地球的一个 UV 贴图

UV 拆分也是一种二次建模, 首先需要定义物体的接缝, 然后才能把其拆开。如图 6-92 左图所示, 要先将一个纸盒用剪刀顺着一条边剪开, 才能将其平铺在桌上, 拆分的效果如图 6-92 右图所示。将这个抽象到 3D 物体的应用中, 就是 UV 拆分, 而其中需要被“剪开”的边, 就是在拆分中做重定义的 Seam (缝合边)。

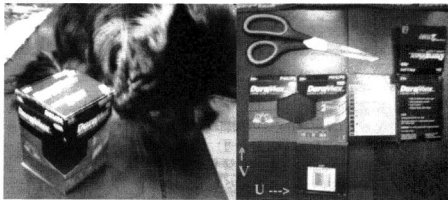


图 6-92 现实中的 UV 拆分

### 6.4.1 UV 坐标系

如图 6-93 所示, 为 UV 坐标系的变换原理图。对比 UV 坐标和 3D 坐标中, 纹理在物体表面的映射效果, 就会发现物体上的纹理比例与 UV 中的贴图存在一定的区别, 这是因为 UV 坐标替

代了 XY 坐标系，而物体上的 Z 坐标深度就必须靠一定的拉伸来表现。所以物体上点的映射位置决定了该点处的拉伸比例，只有合理的映射展开方式才能避免纹理上的过度拉伸。同时如果点面的分布过于密集，也会造成贴图在物体表面产生不正常的变形。

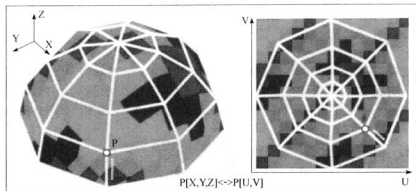


图 6-93 XYZ 坐标与 UV 坐标的转换原理图

UV 常被应用于制作物体表面的纹理色彩，例如皮肤。由于角色的外表会根据物体外形产生不同的变化，简单的映射很容易出现大面积的拉伸，因此在这种情况下推荐使用 UV 来准确定义每个区域的表面特性，而不使用本地坐标系等不容易自定义控制的模式。UV 还适合一些特殊的通道属性应用，例如对于游戏人物等低模模型，可以使用 UV 为其表面精确定位，添加置换和法向等属性来给物体添加高模的细节。如图 6-94 所示，为 Marcin 制作的低模游戏模型 Abandoned Warszawa M20，由于合理地使用了 UV 坐标贴图，并应用了 Diffuse（漫反射）、Normal（法向）和 Specular（高光）等通道属性，模型在面数较低的情况下就可以表现十分丰富而逼真的细节。

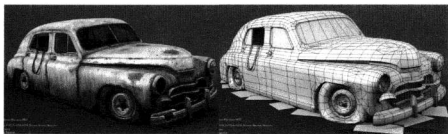


图 6-94 Marcin—Abandoned Warszawa M20

#### 6.4.2 UV 解算和图层管理

拆分 UV 的过程又称为 UV 解算，它是在编辑模式下对物体表面执行的一种重计算过程。当使用快捷键 Ctrl + E 为物体标记好 Seam（缝合边）后，就可以单击快捷键 U 调出 UV Mapping（映射）菜单，选择合适的解算模式，完成坐标的重计算。如图 6-95 所示，为 UV Mapping 解算映射菜单。

**Unwrap：**自动计算最佳的 UV 分布方案，所有被选择的面都将按照缝合边的位置被分离开，这是拆分 UV 时最常用也是最简单快捷的计算方式。如图 6-96 左图所示，为将一个球体对半标记的缝合边，右图则为使用 Unwrap 计

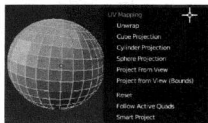


图 6-95 UV Mapping 解算映射菜单

算后的 UV 效果。

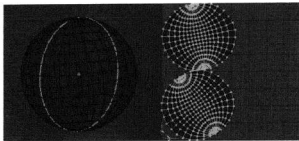


图 6-96 自动计算的 UV 效果

**Cube/Cylinder/Sphere Projection:** 采用立方体/圆柱/球体映射等方式来解算 UV，算法原理同纹理映射类似。如图 6-97 所示，从左至右分别为对同一球体执行三种不同的解算方式的运算效果。



图 6-97 Cube/Cylinder/Sphere Projection（映射解算）的 UV 效果

**Project From View:** 视角映射，系统将根据当前观察物体的视角来映射出 UV 图。如图 6-98 左图所示，为在 3D 视图中观察物体的角度，接着单击选择 Project From View（视角映射），UV 将按照视图的角度分布展开，如图 6-98 右图所示。

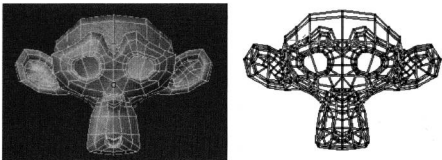


图 6-98 Project From View（视角映射）解算的 UV 效果

**Project From View (Bounds):** 同样采取角度映射，只是这种映射的 UV 结果将采用拉伸模式，自动填满 UV 编辑器的窗口区域，如图 6-99 所示。

**Reset:** 重置所选择物体的 UV 结构。

**Follow Active Quads:** 根据选择激活的边，然后将它们平铺到 UV 编辑器区域。但是由于平铺时系统并不会计算网格的大小，所以解算后需要适当地缩小 UV 以便适应编辑器的窗口区域，

如图 6-100 所示。

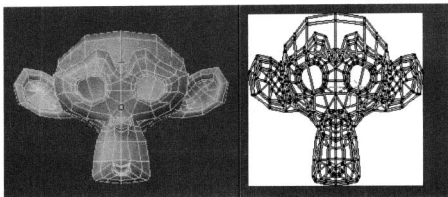


图 6-99 Project From View (Bounds) (自适应角度映射) 解算的 UV 效果

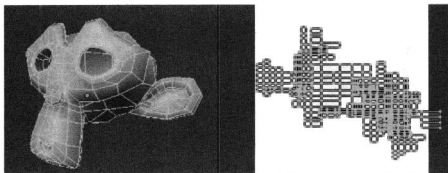


图 6-100 Follow Active Quads (平铺激活面解算) 的 UV 效果

**Smart Project:** 智能解算映射, 此时系统能根据网格上的多边形自动计算 UV, 这个方法适合拆分一些机械模型。如图 6-101 所示, 为对同一个模型采用 Smart Project 解算后的效果。

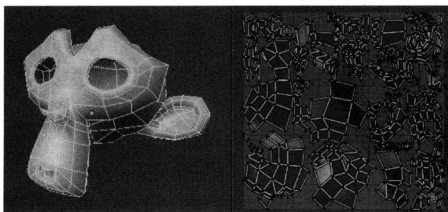


图 6-101 Smart Project (智能解算映射) 的 UV 效果

对于同一个物体, 可以使用多种解算方式来制作一个最合适的 UV 结构。也可以为同一个物体制作多个 UV 解算结构, 只需要在物体的网格属性中添加 UV 组即可分别管理它们。如图 6-102

所示,其中 UVTex 为第一层 UV 结构层,当新建一个 UVTex.001 后,物体的 UV 将被重置,这时就可以使用其他的解算方式来制作一个新的 UV 结构。



图 6-102 多 UV 组管理

### 6.4.3 UV 编辑器

在完成对物体的 UV 解算后,UV 坐标图将自动在 UV Editor (编辑器) 中展开。接下来就需要在 UV Editor (编辑器) 下修改和调整物体的 UV 坐标图,重新排列 UV 上各点和面的分布,以便能充分利用 UV 的贴图区域空间。由于 UV 解算中将不可避免地产生不同程度上的面拉伸,还需要借助 UV Editor (编辑器) 来修改拉伸的幅度,以完善贴图的蒙皮效果。

#### 6.4.3.1 Select (选择) 菜单

在第 2 章中曾简单地介绍了 UV 编辑器的部分菜单功能,如图 6-103 左图所示,在物体模式下 UV 编辑器仅提供部分图片的编辑和预览功能。当进入物体的编辑模式后,UV 编辑器的菜单会增加更多选项,如图 6-103 右图所示,为新增的 Select (选择) 菜单。

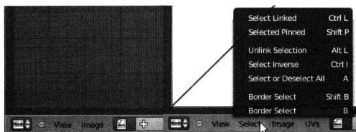


图 6-103 Select (选择) 菜单

**Select Linked:** 关联选择,当存在多个分离的 UV 时,如图 6-104 左图所示,可以首先选择任意 UV 上的一个点,然后单击快捷键 Ctrl + L,系统将自动选择与这个点在同一个 UV 块上的所有面,效果如图 6-104 右图所示。

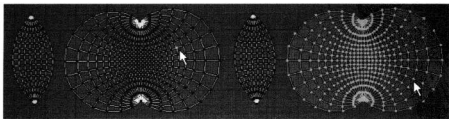


图 6-104 Select Linked (关联选择)

**Selected Pinned:** 选择 UV 上的任意点, 然后单击 P 图钉工具即可将该点钉在编辑区域上, 使其无法被牵连移动, 如图 6-105 左图所示。单击 Shift + P 就可以选择当前 UV 上所有的图钉点, 效果如图 6-105 右图所示。

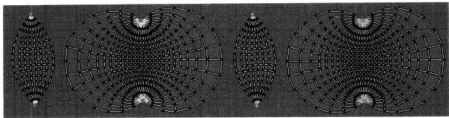


图 6-105 快速选择所有的图钉点

**Unlink Selection:** 仅选择在物体上有相邻关系的 UV 点和面, 快捷键为 Alt + L。

**Border Select:** 框选, 快捷键为 B。

#### 6.4.3.2 UVs 菜单

单击 UVs 菜单, 可以找到一些 UV 编辑相关的工具, 如图 6-106 所示。

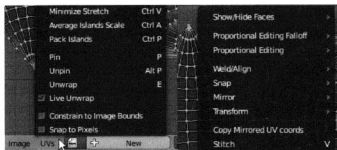


图 6-106 UVs 菜单

**Snap to Pixels:** 将点吸附到编辑区的网格上。

**Constrain to Image Bounds:** 将 UV 的分布约束在编辑区的范围内。

**Live Unwrap:** 使 UV 编辑器与 3D 视图中的修改可以同步刷新, 实现物体实时更新贴图显示的效果。

**Pin/Unpin:** 图钉工具, 快捷键为 P。取消当前选择点的图钉效果, 快捷键为 Alt + P。

**Pack Islands:** 系统对 UV 做自动排列, 使 UV 在编辑区域中的分布可以实现最优化显示, 快捷键为 Ctrl + P。

**Average Islands Scale:** 系统对 UV 做自动缩放, 快捷键为 Ctrl + A。

**Stretch/Minimize Stretch:** 系统将按照相邻面的角度对 UV 做自适应的拉伸缩放调节, 快捷键为 Ctrl + V。如图 6-107 左图所示, 为未拉伸时的效果, 全选 UV 并使用 Ctrl + V 做拉伸变化, 同时配合 MMB 来控制拉伸的强度, 最后效果如图 6-107 右图所示。

**Copy Mirrored UV coords:** 基于物体坐标轴建立对称的 UV 图。

**Transform:** UV 的形变编辑工具, 包括对点或面的 G 移动工具、S 缩放工具和 R 旋转工具, 使用方法和视图中对物体网格的操作类似。

**Mirror:** 镜像工具。

**Snap:** 对齐工具。包括 Selected to Pixels (将选择对象对齐到网格), Selected to Cursor (将选择对象对齐到光标), Selected to Adjacent Unselected (将选择对象对齐到缝合边), Cursor to Pixels

(将光标对齐到网格), Cursor to Selection (将光标对齐到选择物体) 等。UV 编辑器中有一个类似视图中的光标体, 如图 6-108 左图所示, 它常被用于 UV 中的一些定位操作, 例如选择 Selected to Cursor (对齐到光标) 时, 激活的点会被自动移动到光标的位置, 如图 6-108 中图所示。这里最有用的是 Selected to Adjacent Unselected (对齐到缝合边), 它可以两个分离的 UV 块连接到一起, 如图 6-108 右图所示, 点将自动移动到物体表面上和它同位置的点上。

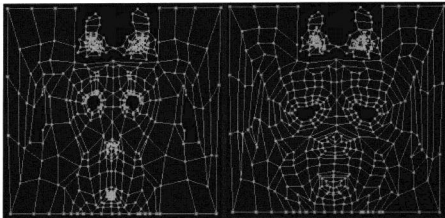


图 6-107 系统对 UV 的自适应拉伸效果

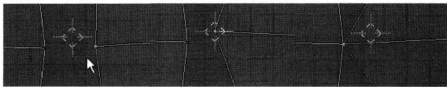


图 6-108 Snap (对齐) 工具

**Weld/Align:** 缝合工具, 当想缝合两个分开物体的 UV 面时, 可以按住 Shift + RMB 选择这些点, 单击快捷键 W 将它们缝合起来。如图 6-109 左图所示, 选择需要被缝合的点后单击 W, 最后的缝合效果如图 6-109 右图所示。

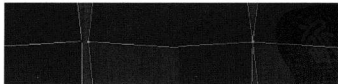


图 6-109 Weld (缝合) 效果

**Proportional Editing:** 开启衰减编辑模式, 和视图中网格的衰减操作方法类似。

**Show/Hide Faces:** 隐藏选中的点和面, 快捷键为 H; 取消隐藏的快捷键为 Alt + H。

#### 6.4.3.3 Display (显示) 菜单

单击 N 可以进入 UV 的属性选项面板, 如图 6-110 所示, 为 Display (显示) 属性的参数面板。

**Normalized:** 使用 0 至 1 的数值范围来显示 UV 编辑区域上的坐标范围, 下面的坐标显示用于识别光标当前的位置。

**UVs:** UV 的显示方式, 如图 6-111 所示, 从左至右分别是: Outline (轮廓)、Dash (网格)、

Black（黑线轮廓）和 White（白线轮廓）模式的显示效果。

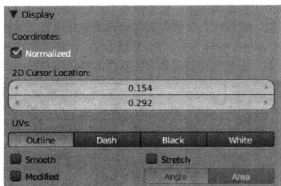


图 6-110 Display（显示）属性的参数面板

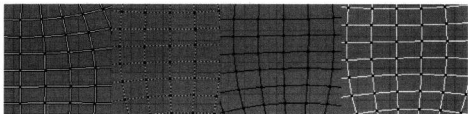


图 6-111 不同的 UVs 显示效果

**Smooth:** 开启 UV 轮廓边缘的平滑显示。

**Modified:** 当为物体添加修改器后，UV 中将显示网格受修改器的作用效果。例如为物体添加一个细分修改器，如图 6-112 左图所示，为未开启 Modified 显示的效果，当开启 Modified 后，UV 中会显示由修改器生成的细分网格边，如图 6-112 右图所示。

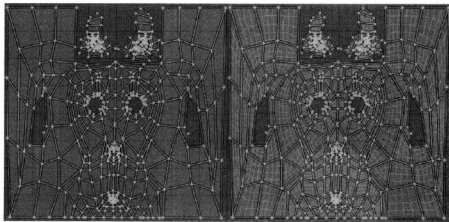


图 6-112 Modified 用于显示物体上修改器的作用效果

**Stretch:** 开启 UV 面的 Z 向拉伸幅度显示，常用于检查 UV 表面的伸展情况。当开启这个选项后，无拉伸的 UV 面将显示为蓝色，如图 6-113 左图所示。如果这个面在 Z 向存在一定的过度拉伸或展开错误，网格会渐变为绿色，在拉伸最严重时会出现红色，效果如图 6-113 右图所示。



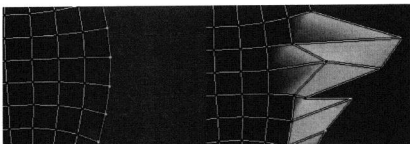


图 6-113 UV 的 Stretch (拉伸) 幅度显示

## 6.4.4 贴图绘制模式

### 6.4.4.1 新建贴图文件

当完成了对 UV 的编辑修改, 就可以添加一个图片来制作 UV 贴图了。可以使用一个现成的图片, 首先单击 Image 菜单, 选择 Open 或使用快捷键 Alt + O 打开一个图片文件。也可以新建一个图片来重新绘制, 单击 Image 菜单, 选择 New (新建), 如图 6-114 左图所示为弹出的新建图片菜单。其中 Width (宽) 和 Height (高) 选项可定义新建贴图的尺寸, color 颜色和 Alpha 值用于控制图片的默认色彩和透明属性。如果选择 UV Test Grid 可生成网格测试贴图, 用于检测物体 UV 拆分后的实际贴图效果, 如图 6-114 右图所示。当选择 32 bit Float 浮点模式时, 可新建一个 32 位的浮点贴图文件。

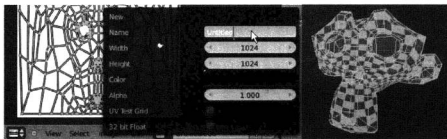


图 6-114 建立一个新的空白贴图

制作贴图时可以使用一些第三方的平面绘制软件, 例如 Photoshop 和 GIMP 等, 也可以直接在 Blender 中进行绘制。首先在新建一个贴图文件后, 单击 Image 选择 Image Painting (贴图绘制), 或者单击标题栏上的画笔图标, 开启颜色绘制功能, 如图 6-115 所示。

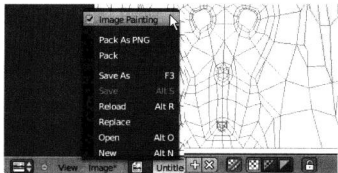


图 6-115 Image Painting (贴图绘制) 模式

## 6.4.4.2 绘制工具

在窗口中单击 N 可调用绘制模式的参数面板, 如图 6-116 所示。

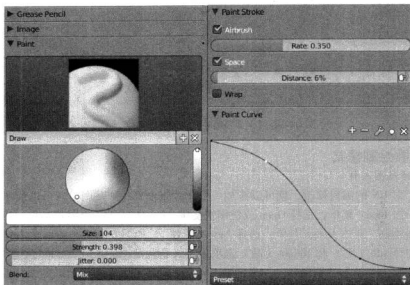


图 6-116 绘图工具面板

**Paint:** 画笔工具, 包括 Draw (绘制)、Soft (柔化)、Clone (克隆) 和 Smear (涂抹) 画笔, 其中 Size (尺寸) 和 Strength (强度) 分别控制笔刷的绘制属性。下面的颜色拾色器可以为笔触选择绘制的色彩, 而 Blend (混合) 模式则可以为绘制选择色彩的融合方式。如图 6-117 所示, 在色板中选择一个颜色后, 单击 LMB 就可以在贴图上绘制了。

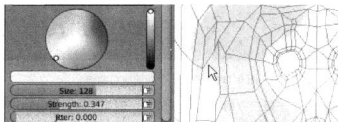


图 6-117 在 UV 上直接绘制贴图的颜色

**Paint Stroke:** 笔触属性, 用于定义是否使用 Airbrush (喷绘) 模式, 以及调节笔触的 Space (间距) 属性。

**Paint Curve:** 笔触控制曲线, 这里可以使用曲线来调整笔刷的边缘形态。可以调整曲线令绘制时笔触的边缘更清晰, 如图 6-118 所示, 为 3D 视图中实时预览的绘制效果。

## 6.4.4.3 Texture Paint (贴图绘制) 模式

当然除了在 UV 编辑器中直接进行贴图绘制, 还可以在模式里选择 Texture Paint (贴图绘制) 模式, 直接在视图中的物体表面上进行绘制。如图 6-119 所示, 当激活这种模式后单击 T 可以调出绘制笔刷面板。

使用 LMB 可在物体上直接进行绘制, 配合使用 RMB 可以在物体上选取色彩值, 如图 6-120 所示。将着色器预览改为 Texture (贴图) 模式, 物体在视图中就会自动显示出 UV 编辑器中的贴图信息, 同时在视图下绘制的色彩也会自动在 UV 编辑器中进行实时更新。

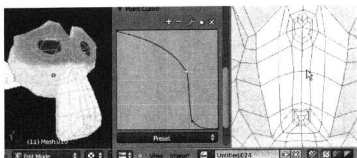


图 6-118 使用曲线来控制笔刷的边缘

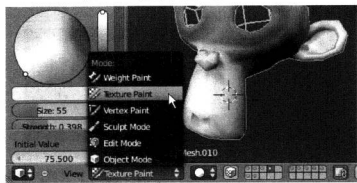


图 6-119 Texture Paint (贴图绘制) 模式



图 6-120 直接在物体上绘制贴图效果

这里还可以为画笔添加纹理，如图 6-121 左图所示。如果开启物体的网格蒙版显示，可以在绘制贴图的时候更方便地对表面做精确定位，效果如图 6-121 右图所示。



图 6-121 为画笔添加纹理绘制

#### 6.4.5 网格 UV 拆分实例

最后，使用一个简单的例子来了解拆分 UV 并制作贴图的全部流程。

首先在 3D 视图中单击 Shift + A，添加一个 Cube 物体，如图 6-122 左图所示。接着单击 Tab 进入其编辑模式，然后选择 Edge (边) 编辑模式，如图 6-122 右图所示。

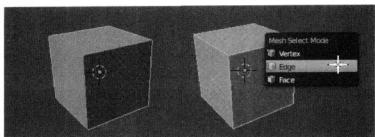


图 6-122 进入新建物体的编辑模式

按住 Shift + RMB 选择物体上的全部接缝边，单击 Ctrl + E 调出边选项菜单，选择 Mark Seam (标记缝合边)，如图 6-123 所示。

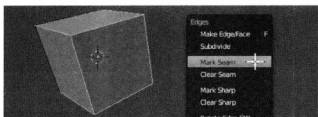


图 6-123 Mark Seam (标记缝合边)

接着在新窗口中切换至 UV/Image 编辑器，单击 Image (图片) 菜单，使用 Open (打开) 一个图片，如图 6-124 左图所示。回到 3D 视图，按 A 选择全部的面，然后单击 U 调出 UV 菜单，单击 Unwrap (解算) 物体 UV，如图 6-124 右图所示。

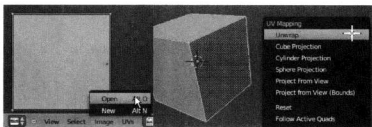


图 6-124 解算 UV

回到 UV 编辑器，就可以看到物体的表面已经被平铺在刚才打开的图片上了，如图 6-125 左图所示。接着使用快捷键 B + RMB 框选 UV 网格点，按住快捷键 G 进行移动等调节操作，同时还可以配合快捷键 S 将 UV 缩放至合适的尺寸。最后的 UV 效果如图 6-125 右图所示，在 3D 视图中，打开 Textured 着色器预览，就可以实时查看贴图在物体上的映射效果了。

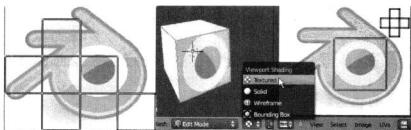


图 6-125 调整 UV 网格

## 第7章

## 渲 染

Rendering（渲染）是将场景中的模型计算成图像的过程，不同的应用领域对渲染的定义也不一样。例如对于VR虚拟现实或者游戏动画，场景和角色都需要实时的显示效果，所以这里的渲染就是指依靠拥有图形加速器的显卡来完成的绘制计算。如图7-1左图所示为Blender开源游戏《Yo Frankie!》的实时渲染效果。而对于电影等高清工业级产品应用，渲染指的是使用大量的光学计算和物理模拟运算的绘制过程，这时系统就需要较强的CPU和大容量内存的支持，甚至使用集群的渲染系统。例如《Elephants Dream》这部时长仅为10分钟的动画电影，其中单帧画面的计算所需内存就高达2.8GB，幸好团队得到了Bowie州立大学的慷慨相助，使得制作人员可以利用其拥有的240台双核CPU的Xserves集群系统，花费了总共125天的时间，才完成了整部电影的渲染。如图7-1右图所示为电影的截图效果。



图7-1 《Yo Frankie!》和《Elephants Dream》的渲染效果

## 7.1 渲染设置

当设置好了场景中物体的材质和环境的灯光之后，可以在标题栏中选择将要使用的渲染器。如图7-2所示，在工具面板中可以找到渲染的参数面板。

Blender Render为系统默认的内置渲染器，它性能十分优秀，提供了高效的光线追踪计算和全局光照计算等。Blender Game为游戏引擎，当制作游戏时可选择其为主渲染器，提供实时的渲染计算。此外Blender还支持网络渲染，选择Network Render后即可进入对应的设置面板。

☐ Image: 图片按钮可渲染静帧图片，其快



图7-2 选择渲染器

快捷键为 F12，当渲染完成后可按 F11 按钮来预览当前的渲染静帧。

- ❑ **Animation:** 动画按钮可渲染出完整的动画连续帧，快捷键为 Ctrl + F12，渲染的过程中可单击快捷键 ESC 来终止渲染，完成后可使用 Ctrl + F11 来回放渲染的动画。

可以实时查看渲染的过程，系统提供了三种 Display（显示）模式。

- ❑ **Full Screen:** 全屏模式，运行渲染后渲染窗口将最大化至全屏显示。
- ❑ **Image Editor:** 图片编辑器模式，这时的渲染结果将在视图中的 Image Editor（图片编辑器）中显示，如果没有激活的 Image Editor（图片编辑器）窗口，Blender 将会自动切换出一个新的 Image Editor（图片编辑器），来显示当前的渲染结果。
- ❑ **New Window:** 新窗口模式，这时系统会弹出一个新的渲染窗口来显示预览画面。如图 7-3 所示为用于查看渲染结果的窗口界面。



图 7-3 渲染窗口的界面

渲染窗口的上方会显示一些常用的渲染信息，例如参与渲染的 Fa 多边形数量，Mem（内存占用）和 Time（渲染时间）等。在渲染窗口中单击快捷键 J 可显示上一次的 Slot 渲染缓存，方便对比和调整。

### 7.1.1 着色选项

在 Shading（着色）选项中，可开启参与渲染计算的着色器相关属性。如图 7-4 所示为 Shading（着色）选项的参数面板。

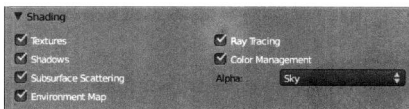


图 7-4 Shading（着色）选项的参数面板

**Textures:** 渲染时是否激活物体的贴图，如图 7-5 左图所示为正常的渲染结果，当取消这个功能时，物体上被赋予的材质贴图都将会隐藏掉，如图 7-5 中图所示。

**Shadows:** 是否开启阴影渲染，如图 7-5 右图所示为关闭这个选项后的渲染，物体在光线下产生的阴影都消失了。

**Subsurface Scattering:** 是否开启 SSS 的渲染计算。

**Environment Map:** 渲染时是否开启对环境贴图的计算。

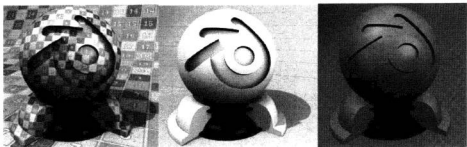


图 7-5 Texture（贴图）和 Shadows（阴影）选项的各种渲染效果

**Ray Tracing:** 是否开启 Ray Tracing（光线追踪）计算，当计算关闭后，物体的相关光线追踪选项都将失效。如图 7-6 左图所示为启动光线追踪实现透明材质的渲染，注意渲染的时间为 1 分 50 秒。右图为关闭光线追踪后的渲染结果，可以发现需要光线追踪来计算的透明效果也被关闭了，但是渲染时间缩短为 6 秒钟。所以，有时在场景要求不高的场合或者在测试渲染中，常常可以关闭 Ray Tracing，以节约渲染的时间。

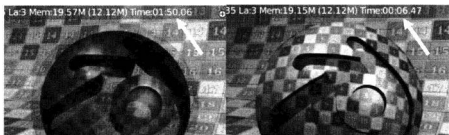


图 7-6 启用关闭 Ray Tracing（光线追踪）功能的渲染效果

**Color Management:** 开启线性化色彩管理工作流。

**Alpha:** 透明背景的填充值，选择 Sky 将使用天空的色彩来填充，如图 7-7 左图所示；选择 Premultiplied 将把物体和背景叠加起来，如图 7-7 中图所示；选择 Straight Alpha 将不显示透明的部分，如图 7-7 右图所示。

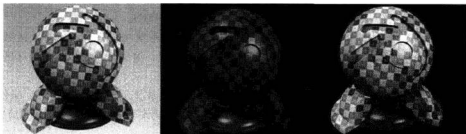


图 7-7 不同 Alpha 值的渲染效果

### 7.1.2 规格选项

在 Dimensions（规格）面板中，可以定义渲染的尺寸等属性。如图 7-8 所示为 Dimensions（规格）的参数面板。可以直接在 Presets（预设值）中选择一些现成的渲染设置，如 HDTV 1080p 或者 TV PAL 16:9 等标准渲染输出。

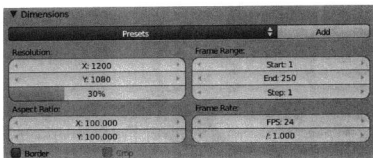


图 7-8 Dimensions (规格) 的参数面板

**Resolution:** X 和 Y 分别定义输出图像宽和高的像素值, 百分比用于调整渲染时采用的尺寸比例。通常使用 25% 或较小的缩放比例来做预览渲染, 使用 100% 做最后的成品输出。

**Aspect Ratio:** 拉伸选项, 其中 X 和 Y 分别控制纵横方向的拉伸幅度。如图 7-9 左图所示为正常比例下的渲染结果, 而右图是 X 是 50、Y 是 100 时的拉伸渲染效果。

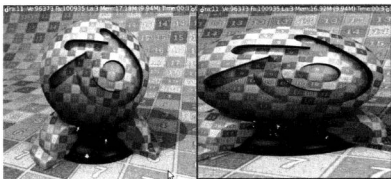


图 7-9 Aspect Ratio (拉伸) 效果

**Frame Range:** 在制作或渲染动画时, 可以在这里设置动画的 Start (起始) 帧位置, 和动画的 End (结束) 帧位置, 以及 Step (单位步进) 的帧数。

**Frame Rate:** FPS (Frames Per Second) 每秒钟运动图像的帧数选项, 动画中每秒钟的帧数越多, 人物角色的动作就越流畅, 对于普通动画一般设置为 24FPS。

**Border/Crop:** 设置一个自定义的渲染区域, 这样渲染时系统将仅计算 Border (区域) 范围内的图像, 快捷键为 Shift + B。如果开启了 Crop (剪切) 功能, 渲染时将根据 Border (区域) 自动剪切渲染的画面。

由于 PC 上图形的显示像素为正方形, 所以要在 PC 上正常显示渲染效果的话, 应保持 Aspect 默认为 100/100。但是对于特殊的电视和电影制式, 就需要对输出比例和帧率做相应的调节。如表 7-1 所示为常用的视频输出格式规范。

表 7-1 常用视频制式规范

Standard 制式	Resolution 分辨率	Aspect Ratio 拉伸比	Frame Rate 帧率
HDTV 1080p	1920 × 1080	1:1	24
HDTV 720p	1280 × 720	1:1	24
TV NTSC	720 × 480	10: 11	30
TV PAL	720 × 576	54: 51	25
TV PAL 16:9	720 × 576	64: 45	25



### 7.1.3 运动模糊选项

真实世界中的摄像机拥有快门速度和曝光时间等物理属性，所以在拍摄相邻的两帧画面之间会产生一定的曝光延时。如果拍摄的是高速运动的物体，画面中的物体就会出现一种运动模糊的效果。在动画中可以使用 Motion Blur（运动模糊）计算来模拟实际摄像机的这种工作模式，以增强动画的真实感。如图 7-10 所示为 Full Sample Motion Blur（运动模糊采样）的参数面板。



图 7-10 Full Sample Motion Blur（运动模糊采样）的参数面板

**Motion Samples:** 采样率的值决定了模糊的融合度，值越大物体拖动出的动态阴影越平滑。让球体从画面的左方向右方运动，如图 7-11 左图所示为设置 Motion Samples 值是 3 的运动模糊效果，可以看到物体的拖影存在“洋葱皮现象”。如图 7-11 右图所示为将 Motion Samples 增加到 8 时的渲染效果，这时的物体运动模糊就变得更平滑，效果也更自然。

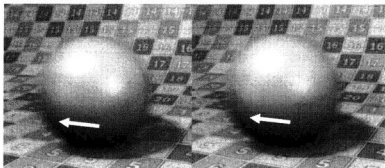


图 7-11 Full Sample Motion Blur（运动模糊采样）的效果

**Shutter:** 快门时间的控制参数，一般保持默认即可。

### 7.1.4 抗锯齿

计算机显示的图像都是由像素组成的，渲染器会在计算时为每个像素点分配颜色信息。但是由于分辨率对像素的排列数量的限制，在较低分辨率下的物体边缘就会因为缺少足够的色彩信息而产生一定的凹凸锯齿效果。Anti-Aliasing 简称 AA（抗锯齿），它是一种用于消除渲染结果中物体边缘出现凹凸锯齿现象的计算功能。如图 7-12 所示为 Anti-Aliasing（抗锯齿）的参数面板。



图 7-12 Anti-Aliasing（抗锯齿）的参数面板

**Sample:** 较高的 Sample（采样率）可以使画面在渲染中得到更平滑的物体边缘效果，但渲染时间也会相应增加。Blender 的内置渲染器在每个像素上使用采样点来过滤数据，因此采样率越高每个像素点上的采样点也就越多。渲染时采样点可将这些像素点的色彩值做混合处理，使各像素点上都具有临近像素点的色彩特征，这样像素点之间的色彩就会产生一定的过渡效果，使整个图形的边缘色彩区域看上去更平滑。如图 7-13 所示分别为 5、8、11 和 16 倍采样率的混合模型，其中 Size（尺寸）值用来控制采样点和像素值的分布比例，因此 Sample 值越高，像素上可混合的采样点个数越多，抗锯齿的效果也就越好。



图 7-13 Sample (采样率) 模型

Filter (过滤器) 可以辅助渲染器在采样时对采样的像素做筛选处理。不同的算法决定了每个采样点如何去挑选参与混合的色彩信息, 这里一共提供了 7 种过滤器。

- ☐ **Box:** 矩形波过滤, 它的效果相对于其他过滤器来说最清晰, 而且速度较快。
- ☐ **Tent:** 三角波过滤, 能提供一个很简单的尖锐边缘。
- ☐ **Quadratic:** 二次过滤。
- ☐ **Cubic:** 基于立方体线条的立方过滤。
- ☐ **Catmull-Rom:** 只读存储, 是一种具有显著边缘增强效果的过滤器。
- ☐ **Gaussian:** 高斯过滤, 效果相对比较模糊。
- ☐ **Mitchell-Netravali:** 米切尔过滤器, 是效果较好的过滤器。

如图 7-14 左图所示为关闭 AA (抗锯齿) 的渲染结果, 而右图为开启抗锯齿的渲染效果。对比之后可以看到, 经 AA 处理的图像边缘显得更平滑了, 但是渲染时间却增加了数倍。



图 7-14 AA 抗锯齿渲染的效果对比

### 7.1.5 烘焙

Bake (烘焙) 是将模型的材质和网格等属性, 包括纹理和表面凹凸等信息, 结合场景中的光照数据, 制作成一种特殊 2D 纹理贴图的过程。如图 7-15 所示为 Bake (烘焙) 的参数面板。

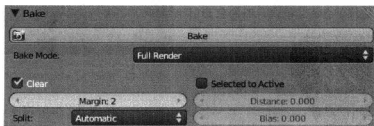


图 7-15 Bake (烘焙) 的参数面板

烘焙是一种预计算的过程, 经烘焙后生成的 2D 贴图可以存储一些渲染时才会产生的数据, 因此合理应用烘焙贴图能加快最终的渲染性能。烘焙贴图常被应用于动画和游戏制作中, 可以有效地节约大量的渲染计算时间。

**Clear:** 烘焙前清除上一次烘焙区域的贴图。

**Margin:** 控制烘焙纹理的边缘粗细，这里的数值用于调整线条像素。

**Split:** 烘焙时四边形将被拆散为三角形，方便计算。这里的选项用于控制网格的拆分模式，默认为 Automatic 自动。

**Selected to Active:** 是否烘焙当前选择的物体对象。其中 Distance 用于控制被选择物体的烘焙影响范围，当相邻的物体与烘焙物体的距离小于当前设置的 Distance 值时，系统将会计算间距内物体对烘焙效果的影响。Bias（偏移量）可使与烘焙对象表面具有一定偏移量的物体也参与到当前的计算中，例如使用烘焙来制作衣服表面上的徽章阴影。

烘焙的工作流程是，被烘焙的对象首先必须拆分好 UV，接着在烘焙面板中选择对应的烘焙模式。如果使用的是 Full Render 模式，系统将烘焙所有通道中的贴图等信息。可以在模式中选择特定的烘焙模式。

**Ambient Occlusion:** 忽略环境中的灯光效果，将物体上接受到的 AO 环境光烘焙至贴图文件中。可以将所有物体的 AO 计算都预先烘焙至贴图，并映射至材质的环境光照通道中，这样在渲染时即可关闭 AO 的计算来节约渲染时间。如图 7-16 左图所示为烘焙的大圣 AO 贴图，如图 7-16 右图所示为将 AO 贴图加在物体上做实时显示的效果。

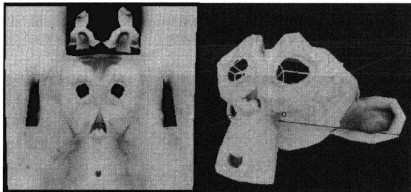


图 7-16 AO 烘焙的贴图效果

**Shadow:** 将由其他物体在自身表面上产生的阴影烘焙至贴图文件中，常用于游戏场景中，这样可以省去灯光的计算而直接生成阴影效果。如图 7-17 左图所示，在一个 Plane 物体上打上灯光后执行烘焙计算，其中大圣在物体上的阴影以及灯光的照明范围都将被完全烘焙至贴图中。如图 7-17 右图所示为将最后得到的阴影烘焙贴图，而此时即使没有灯光，物体的表面也可以出现阴影的效果。

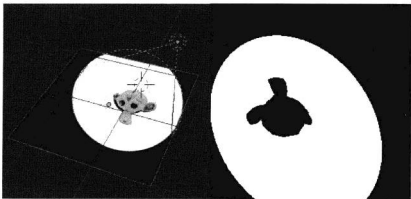


图 7-17 Shadow（阴影）烘焙的贴图效果

**Normals:** 烘焙法线贴图。首先需要制作一个分好 UV 的物体, 如图 7-18 左图所示。接着在物体模式下, 单击 Shift + D 复制一个新物体出来, 并将其制作为高模物体, 如图 7-18 右图所示。

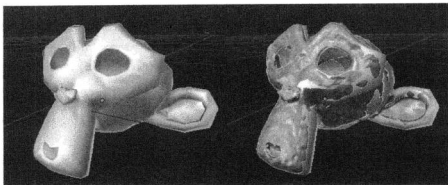


图 7-18 分别制作低模和高模物体

选择低模物体, 在编辑模式下为其新建一个贴图文件。接着回到物体模式中, 再选择高模物体, 然后单击 Shift + LMB 同时选择低模物体, 并启动 Bake (烘焙) 程序。如图 7-19 左图所示为烘焙完成的法线贴图, 将其应用至低模物体的法线通道上, 如图 7-19 右图所示, 这样就可以在低模物体上表现出高模细节了。



图 7-19 Normal (法线) 烘焙的贴图效果

**Textures:** 烘焙纹理贴图, 将材质上的纹理色彩烘焙成一个按照 UV 展开的贴图文件。

**Displacement:** 和 Normal (法线) 贴图一样, Blender 也可以烘焙置换贴图, 烘焙方法同法线烘焙。

### 7.1.6 命令行渲染

除了直接在 Blender 界面下调整参数单击按钮执行渲染外, 还可以使用命令行或脚本工具, 在不启动 Blender 界面的情况下直接执行渲染操作。这样做不仅可以节约较多的系统资源, 同时还适合执行批处理操作或网络集群渲染。

首先启用控制台命令行, Blender 的命令行的使用格式如下所示:

```
>blender[args...][file][args...]
```

也可以直接在 Blender 的根目录下执行如下代码, 获得 Blender 的具体参数帮助信息:

```
>Blender.exe/?
```

### 7.1.6.1 Render (渲染) 参数

- b 或者 --background <file>: 载入参与渲染的 file 工程文件。
- a 或者 --render-anim: 开启设置好起始帧和结束帧的动画工程。
- s 或者 --scene <name>: 选择将要渲染的 Scene 场景名。
- f 或者 --render-frame <frame>: 渲染指定帧数的画面, 不可以和参数 -a 一起使用。
- s 或者 --frame-start <frame>: 设置渲染的起始帧数, 使用时应设置在参数 -a 之前。
- e 或者 --frame-end <frame>: 设置渲染的结束帧数, 同理也应设置在参数 -a 之前。
- j 或者 --frame-jump <frames>: 设置渲染动画的步进帧。
- o 或者 --render-output <path>: 设置渲染的输出路径和输出文件名, 使用//来设置文件存储的相对路径, 在文件名后添加#可标识渲染的帧数。例如设置输出为 ani\_##\_test.png, 那么最后的渲染输出文件将被命名为 ani\_01\_test.png。
- E 或者 --engine <engine>: 用于选择渲染器, 使用 -E help 命令可显示所有可用的渲染器。

### 7.1.6.2 Format (输出格式) 参数

- f 或者 --render-format <format>: 设置输出的文件格式, Blender 支持的格式包括 TGA, IRIS, HAMX, FTYPE, JPEG, MOVIE, IRIZ, RAWTGA, AVIRAW, AVIJPEG, PNG, BMP, FRAMESERVER, HDR, TIFF, EXR, MPEG, AVICODEC, QUICKTIME, CINEON 和 DPX 等。
- x 或者 --use-extension <bool>: 是否添加文件后缀名, 0 为否, 1 为是。
- t 或者 --threads <threads>: 设置渲染调用的 CPU 进程数, 使用 [1 - BLENDER\_MAX\_THREADS], 0 将自动调用系统的所有 CPU 可用进程。

### 7.1.6.3 Animation Playback (动画回放) 参数

- a <options> <file (s)>: 设置回放的 file (文件) 名。
- f <fps> <fps-base>: 自定义回放的 FPS 帧率。
- p <sx> <sy>: 约束画面的尺寸, 在 Windows 系统下, 这个命令无效。
- m: 不缓存, 直接从硬盘读取文件。

### 7.1.6.4 Window (窗口) 参数

- w 或者 --window-border: 开启窗口的边框显示, 这也是默认设置。
- w 或者 --window-borderless: 默认不使用窗口边框, 只针对 Linux/Unix 系统有效。
- p 或者 --window-geometry <sx> <sy> <w> <h>: 约束窗口的尺寸。

### 7.1.6.5 Game Engine (游戏引擎) 参数

- g fixedtime: 使用固定的 50 赫兹刷新率。
- g vertexarrays: 使用点阵列渲染模式, 速度较快。
- g noaudio: 关闭音频效果。
- g nomipmap: 关闭纹理的 Mipmapping (贴图) 效果。
- g linearmipmap: 使用线性 Mipmapping (贴图) 效果。

### 7.1.6.6 Misc (杂项) 参数

- d: 开启 Debug 功能, 系统将打印调用的控制器名称和其参数输出。
- debug-fpe: 开启浮点指针的异常处理。
- noglsl: 关闭 GLSL 着色计算。
- noaudio: 关闭音频。
- setaudio: 开启音频效果。
- h 或者 --help: 打印当前命令的帮助信息。

- y 或者 --enable-autoexec - : 开启脚本的链接调用, 为系统的默认设置。
- Y 或者 --disable-autoexec: 关闭链接的脚本, 例如关闭使用脚本制作的驱动器或约束控制等。
- P 或者 --python <filename> : 运行指定的脚本名。
- R: 添加 .blend 文件后缀, 仅适用于 Windows 系统。
- v 或者 --version: 打印 Blender 版本后退出。

#### 7.1.6.7 参数应用法则与范例

如果希望渲染一个单帧画面, 可直接在命令行中输入以下命令:

```
>blender -b file.blend -o//file -F JPEG -x 1 -f 1
```

其中, -b file.blend 为使用的渲染工程文件, -o//file 指定渲染输出路径和文件名, -F JPEG 设置输出为 JPEG 的图片格式, -x 1 添加了文件后缀名, -f 1 控制渲染第 1 帧的画面。

当渲染一个动画时, 可调整命令行如下:

```
>blender -b file.blend -x 1 -o//file -F MOVIE -s 003 -e 005 -a
```

其中, -b file.blend 为使用的渲染工程文件, -x 1 设置文件后缀名为 .avi, -o//file 指定渲染输出路径和文件名, -F MOVIE 指定输出为 AVI 视频格式, -s 003 -e 005 -a 则设置渲染的起始帧为 003, 结束帧为 005。

在调用命令行时, 还需要掌握一些基本的参数规范, 例如参数之间必须使用空格来间隔, 如果将命令书写成了:

```
>blender -ba test.blend
```

那么系统将会忽略掉输入的参数 a, 而改为执行如下命令:

```
>blender -b test.blend
```

同时, 参数的顺序也会影响执行的效果, 例如执行如下代码:

```
>blender --background test.blend --render-frame 1 --render-output/tmp
```

当单击回车键执行的时候就会发现, 渲染并不会将结果输出至目录/tmp 中, 这是因为命令 --render-frame 1 执行完成后, 渲染就已经结束了, 因此路径设置应放置在最终渲染命令之前。

但是如果设置命令如下:

```
>blender --background --render-output/tmp test.blend --render-frame 1
```

系统仍然不会将结果输出到目录/tmp 中。这是因为在完成路径设置后, 又命令系统重新调入 test 文件, 这样就会造成原来的设置被新导入的文件给覆盖掉。所以如果要正确完成渲染, 应修改命令如下:

```
>blender --background test.blend --render-output/tmp --render-frame 1
```

## 7.2 输出设置

### 7.2.1 格式选项

Output (输出) 设置包含渲染的输出格式等参数。如图 7-20 所示为 Output (输出) 的参数面板。

在这里可以为渲染图片或视频指定输出的路径, 其中/tmp \ 默认为安装路径下的 tmp \ 文件夹。也可以单击旁边的文件夹小图标, 在新弹出的窗口中选择自定义的路径来存储渲染输出。

Blender 支持多种格式输出, 默认输出为

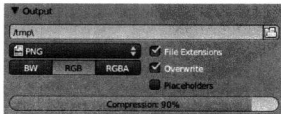


图 7-20 Output (输出) 的参数面板

PNG。如图 7-21 所示,单击格式按钮可弹出 Format (输出格式) 菜单。

### 7.2.1.1 Image (图片类) 格式

Blender 不仅支持常用的 PNG、JPEG 和 TGA 等文件格式,还支持适合后期输出合成、带 Z 缓冲通道的 32 位浮点精度 OpenEXR 等文件格式。

**PNG:** Portable Network Graphics, 即可移植性网络图像,这是一种比 GIF 更优化的网络图形文件,它压缩比率小,支持 24 位和 48 位真彩色图像和 Alpha 通道。

**JPEG/JPEG 2000:** Joint Photographic Experts Group, 即联合图像专家组,这是最常用的

图像文件格式。它是一种有损压缩文件格式,在去掉图像中重复和不重要的数据后,还能展现十分丰富生动的图像。JPEG 文件支持多种压缩级别,压缩比率通常在 10:1 ~ 40:1 之间,其中压缩比越大,品质越低。JPEG 2000 是 JPEG 的升级版,压缩比率比 JPEG 高约 30%,支持有损和无损两种压缩方式。JPEG 2000 最重要的功能就是它支持网络的渐进传输,即先传输图像的轮廓,然后逐步传输数据,不断提高图像质量,让图像由朦胧到清晰。此外, JPEG 2000 还支持所谓的“感兴趣区域”特性,可以指定画面上任意区域的压缩质量,或者让选择的部分先执行解压缩。

**BMP:** Bitmap 是最早使用的 Windows 图像文件格式,它采用位映射的存储格式,但是由于不采用任何压缩计算,所以 BMP 文件所占用的空间很大。

**Targa/Targa Raw:** Truevision Advanced Raster Graphic Adapter 是美国 Truevision 公司为其显卡开发的一种图像文件格式,最早在 IBM 的 PC 系统上使用。Tga 文件架构比较简单,属于一种图形数据的通用格式,它支持压缩,而且使用的是不失真的压缩算法,因此是一种比较好的图片格式。

**HamX:** Blender 自行开发的 8 位 RLE (Run Length Encoded Bitmap) 文件格式,常用于动画预览时的内部调用。

**Iris:** 是标准 SGI (Silicon Graphics Inc) 文件格式,常用于 Unix 系统。

**OpenEXR:** OpenEXR 由 ILM (Industrial Light and Magic, 工业光魔特效) 公司开发,是一种高动态范围影像的文件格式。这种格式具有较高的色彩真实度和动态范围,能满足影视级电影的制作需求。

**Multilayer:** 多图层格式,它也采用了 OpenEXR 的格式,并支持多图层单文件的存储格式,每个图层包括一个渲染通道,例如阴影、高光 and 色彩等。

**TIFF:** Tag Image File Format 是由 Aldus 和 Microsoft 为桌面出版系统研发的一种通用图像文件格式,常用于传真和通信。

**Radiance HDR:** 这是一种 High Dynamic Range (高动态图像) 格式,它保存了足够的光照信息,拥有比普通 RGB 格式图像更大的亮度范围,记录了远远超出 256 个级别的实际场景的亮度值,其中超出的部分在屏幕上显示不出来的。

**Cineon:** 由 Kodak 公司开发的一种适合电子复合的 10 位通道数字格式, Cineon 可以在不损失图像品质的情况下输出胶片,常用于 Cineon Digital Film System (电影数码系统)。

**DPX:** Digital Moving-Picture eXchange 文件格式的缩写,这是一种开放式的专业图形格式,和 Cineon 类似,常用于影视数码系统制作。



图 7-21 Format (输出格式) 菜单

### 7.2.1.2 Movie (视频类) 格式

视频类格式包括无压缩的 AVI 格式和 FFMpeg 等压缩编码格式。当选择不同的输出格式时,面板会弹出对应的编码参数设置。

**AVI Raw/Jpeg/Codec:** AVI 是 Audio Video Interleave 视频音频交错文件格式的缩写, 是由 Microsoft 公司开发的一种标准视频格式。所谓“音频视频交错”, 就是指它可以将视频和音频交织在一起进行同步播放。这种视频格式的优点是图像质量好, 可以跨多个平台使用, 其缺点是体积过于庞大, 而且更加糟糕的是压缩标准不统一。最普遍的现象就是高版本 Windows 媒体播放器播放不了采用早期编码编辑的 AVI 格式视频, 而低版本 Windows 媒体播放器又播放不了采用最新编码编辑的 AVI 格式视频。所以在输出一些 AVI 格式的视频时, 常常由于视频编码问题, 出现视频不能播放, 或者即使能够播放, 但不能调节播放进度, 甚至只有声音没有图像的情况。如果在进行 AVI 格式的视频播放时遇到了这些问题, 可以尝试通过下载相应的解码器来解决。这里提供的 AVI Jpeg 可将视频按帧渲染, 每一帧都以 Jpeg 格式存储起来, 文件体积会相对缩小。AVI Codec 为视频提供压缩式存储, 压缩比远小于 AVI Jpeg 格式。

**QuickTime:** 是 Apple (苹果公司), 开发的一种视频格式, 默认的播放器是 Quick Time Player。MOV 文件具有较高的压缩比率和较完美的视频清晰度等特点, 其最大的特点是跨平台应用, 它不仅能支持 Mac OS 系统, 同样还能支持 Windows 系统和 Linux 系统。

**H. 264:** 这是由 JVT (Joint Video Team, 视频联合工作组) 推出的一种为了配合蓝光 HD 视频的高性能视频编码规范, 在同等画质的基础上, 它的压缩比是 MPEG-2 的 2 倍以上, 是 MPEG-4 压缩比的 1.5~2 倍。

**Xvid:** 是由原 OpenDivx 的义务开发者在 OpenDivx 停止开发后编写的一款开源 MPEG-4 视频编码标准。

**Ogg Theora:** Ogg 全称是 OGGVobis, 它是由 Theora 基金会开发的开源有损影像压缩编码技术 Theora 影像流存储, 本身也是一种开放版权的多媒体格式。

**FFMpeg:** Fast Forward MPEG 的缩写, 是一种开源的视频音频压缩编码库, 其编码质量很高, 是视频输出的优选格式之一。

**Frame Server:** 是一种应用于网络渲染农场的文件格式。

Output 输出面板中还有以下几个附加选项。

**BW/RGB/RGBA:** 黑白/彩色和带 Alpha 通道的彩色文件格式。

**File Extensions:** 是否为文件自动添加后缀名。

**Overwrite:** 是否自动覆盖渲染结果。

**Placeholders:** 创建一个空文档来存储渲染输出。

## 7.2.2 性能选项

如图 7-22 所示为 Performance (性能) 选项的参数面板。

**Threads:** 线程数, Blender 的渲染支持多核多线程的 CPU, 当选择 Auto-detect (自动探测) 时, 系统将自动使用当前可用的所有 CPU 线程资源。如果选择 Fixed (固定线程) 选项, 下面的 Threads 会自动激活, 为系统自定义可分配的 CPU 资源。

**Tiles:** 渲染时系统可以将画面按照 X×Y 的比例分割开来, 例如当 X=Y=8 时, 图片将被分割为 64 块 (8×8=64), 每一块分别由一个线程来单独计算, 这样可以有效地利用多线程计算, 提高渲染的效率和资源分配。

**Memory:** 系统的内存是有限的, 渲染和后期节点处理都需要消耗较多的内存。Save Buffers (保存缓存) 模式可将渲染图层和场景节点在 temp 目录下保存为临时缓冲文件, 以便节约内存。

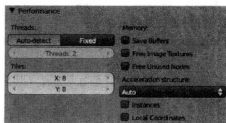


图 7-22 Performance (性能) 选项的参数面板



Free Image Textures 可清空渲染时载入到内存中的贴图数据,为后期的合成提供更多的可用内存,而 Free Unused Nodes 则可清空后期合成时未使用的结点所占用的内存。

**Acceleration Structure:** 光线追踪的加速模式选择菜单,一般保持为 Auto (自动) 模式。

**Local Coordinates:** 将点坐标转换为本地坐标系,这样可能会增加内存的消耗,但是可在一定程度上加快渲染速度。

### 7.2.3 后期合成选项

在渲染计算结束后,可以开启结点编辑或序列编辑等后期合成模式,来对渲染输出做最后的一些调整。如图 7-23 所示为 Post Processing (后期合成) 的参数面板。



图 7-23 Post Processing (后期合成) 的参数面板

**Compositing:** 渲染后可将输出结果自动导入到 Node (结点) 编辑器中,并开启结点计算功能。

**Sequencer:** 开启后,渲染将自动转为执行序列编辑器中的视频输出。

**Fields:** 场选项。电视中画面的扫描时间是根据交流电源的频率来决定的,例如我国的交流电频率为 50Hz,那么显示的画面扫描帧就是 50 帧/秒,但是这样只能达到 288P 的扫描行水平,无法满足人眼的视觉要求,因此电视机就利用隔行扫描的方式,使得画面的显示时间可以提高至 25 帧/秒。但是问题就在于,这样看到的画面只有原画面的一半,因此电视的信号源,也就是电视台会制作具有场信号的电视节目,将每个画面分割成两个部分,通过隔行显示的方式记录每一帧的画面,使得信号到达电视时就可以还原为正常的播放效果。所以如果要制作电视应用的画面,就需要设置场控制,通过 Upper First (上场优先) 和 Lower First (下场优先) 来控制扫描的方向。当然没有场的图像在电视上也可以正常播放,只是这样的画面如果出现快速运动的物体时,就会出现剧烈的跳帧现象。在我国只有使用 Upper First (上场优先) 模式才能保证信号的正常播放。

**Dither:** 为画面添加抖动的杂色。

**Edge:** 线条渲染模式,开启后物体在渲染时会加粗物体外形的线条边缘,这常用于卡通渲染,添加明显的物体轮廓。Threshold 控制线条的粗细,值越大线条越宽。如图 7-24 左图所示是 Threshold 为 10 时的渲染效果,当增大 Threshold 到 25 时,渲染效果如图 7-24 右图所示。

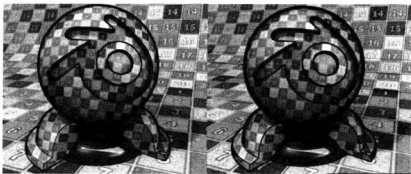


图 7-24 Edge (线条) 渲染效果

## 7.2.4 标记选项

单击 Stamp 菜单选项后, 可以为渲染结果添加自定义的标记文字。如图 7-25 所示为 Stamp (标记) 选项的参数面板。

各个选项分别表示可标记的内容, 这些标记能提供很多有效的信息, 在一些测试渲染或动画的 layout (线稿) 输出中非常有用。如图 7-26 所示为单击渲染后的 Stamp 效果。

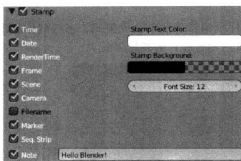


图 7-25 Stamp (标记) 选项的参数面板



图 7-26 Stamp 信息输出效果

## 7.3 层输出属性

系统在执行渲染时, 计算机的硬件性能和工程的参数设置决定了渲染需要花费的时间。为了能按时完成预期的工作, 可以尝试使用分层渲染和通道渲染, 将一个场景分为几部分来分别进行渲染。这样做不仅可以减轻一次性渲染对机器的压力, 而且还便于对每个对象的属性进行单独调整和修改。

### 7.3.1 分层渲染

前面曾介绍了 Layers (图层) 的用法, 它可以用来分别存储和管理场景中不同的物体。同理, 在渲染中也可以指定渲染器来渲染特定的图层。

#### 7.3.1.1 层管理

如图 7-27 所示为 Layers (图层) 的参数面板。

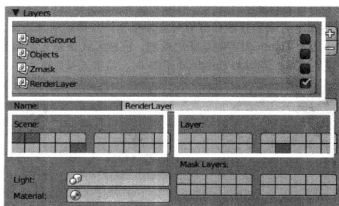


图 7-27 Layers (图层) 参数面板

**Render Layer:** 渲染输出图层, 在面板最上方的层管理框中, 可添加或者删除一个 Render Layer (渲染图层), 使用 Name 选项分别为其命名。每一个渲染图层都是一个单独的渲染输出,

它们可以直接导入到结点系统中，也可以保存为文件。默认情况下，系统只有一个渲染图层。

**Scene:** 场景层，用于选择在视图中显示的层，当激活某一层时，该层中的物体会相应出现在视图场景中。可以单击 Shift + LMB 来选择多个层，操作方式和视图窗口中对层的管理模式类似。

**Layer:** 渲染层，只有被激活的渲染层才能参与最后的渲染计算。这里要注意的是，对渲染层的操作不会影响场景层中物体的显示，它只是控制渲染时参与运算的层。而 Scene（场景）层则存储了可参与渲染的所有层，因此被激活的场景层不一定会渲染出来，但是一个没有被激活的场景层，即使在渲染层中被激活也一定不会被渲染出来。

**Light:** 渲染灯光组，使当前渲染输出层中的所有物体仅被指定的灯光组照亮。通常在测试渲染中会为场景制作好几套灯光设计组，将灯光编组后使用 Light 选项，就可以自定义照明的测试渲染了。

**Material:** 在渲染场景时，使用一个特定材质来覆盖全部物体的材质属性，这在快速调整灯光和特殊场景设计时十分有效。

除此之外，层面板还提供了渲染输出层的属性控制，如图 7-28 所示为输出层的属性选项。



图 7-28 渲染输出层的属性选项

**All-Z:** 计算所有的 Z 通道属性值，并直接叠加到渲染结果中。

**Solid:** 开启物体表面属性的渲染输出，其中物体的表面包括所有带法线的矢量面。

**Halo:** 开启 Halo 的材质渲染。

**ZTransp:** 开启透明追踪属性的渲染计算。

**Sky:** 开启场景中天空在渲染时对物体环境的影响效果。

**Edge:** 开启物体的边渲染。

**Strand:** 开启毛发等粒子物体的 Strand 属性渲染。

#### 7.3.1.2 分层输出

接下来以实例来讲解层在渲染中的应用。首先在场景中将背景物体放在场景层 1，物体放在层 6，灯光放置在层 10。如图 7-29 所示，分别建立两个渲染输出图层，第一个图层命名为 BackGround，设置渲染层为层 1 和层 10。

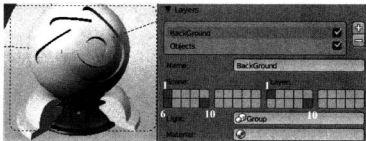


图 7-29 BackGround（背景层）

接着建立第二个渲染输出图层 Objects，设置渲染层为层 6 和层 10，如图 7-30 所示。

这样，在结点系统中就可以使用两个输入结点来分别调用渲染的输出层，为后期合成提供

输入源。如图 7-31 所示为节点系统中的合成示例图。可以看到场景被分为了两个输出层来渲染，其中层之间可使用 Alpha 透明通道的方式来完成合并操作。分层渲染的优势在于，如果需要调整画面中的某一个物体，将不需要再花时间去渲染其他不需要改动的物体和背景层，只需要单独渲染存放修改物体的那一层即可。例如如果要修改物体层的材质，那么渲染时就没有必要再重复渲染一次背景层。同时，分层渲染也有利于合成输出，特别是结点的处理合成。具体应用将在后面的第 10 章中详细讨论。

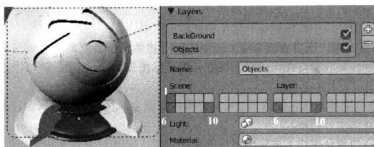


图 7-30 Objects (物体层)



图 7-31 使用结点合成制作分层的输出结果

### 7.3.1.3 Zmask (遮罩层)

这是一种特殊的 Z 深度遮罩式渲染层，常用于有遮挡出现的分层渲染输出。例如，如图 7-32 所示的一个场景，在层 1 中添加一些物体来遮挡层 2 中的球体。

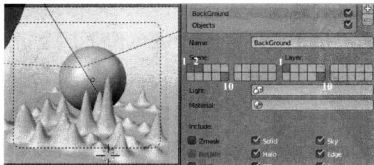


图 7-32 有相互遮挡的层场景

单击渲染后，会发现 Object 层中的物体会被 BackGround 层中新建的物体遮挡，使得这一部分无法渲染出来，而且 Object 层中的 Alpha 通道仅显示物体与环境的轮廓线，造成合成中出现错位现象，如图 7-33 所示。

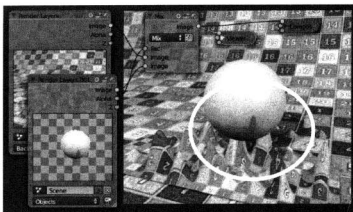


图 7-33 遮挡物体在分层渲染后的错位问题

那么这个时候就需要添加第3个渲染输出图层 Zmask, 如图 7-34 所示, 激活下面的 Zmask 选项后, 面板中会弹出一个 Zmask Layers 层管理区。将新建的遮挡物体放置在场景层 2 中, 在输出层中激活放置被遮挡物体的输出层 1, 由于遮挡物对球体产生了遮挡, 所以在 Zmask 层中激活放置遮挡物体的遮罩层 2。

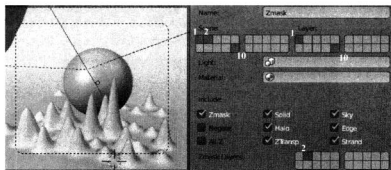


图 7-34 添加 Z-mask 控制层

单击渲染, 可以看到物体上产生重叠的部分就被单独渲染出来了。进入结点编辑器中, 将刚才新建的遮罩渲染输出层导入, 并将它叠加到原来的渲染效果中, 如图 7-35 所示, 这样就解决了输出合成中由于使用分层渲染而出现的遮挡问题了。

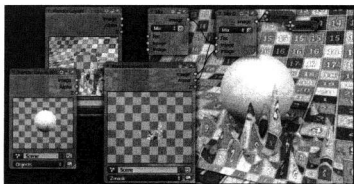


图 7-35 使用 Zmask 修正了图层间的遮挡问题

**Negate:** 配合 Zmask 使用, 用于渲染被遮挡物体背后的场景部分。如图 7-36 所示为激活后 Zmask 层渲染出背景层被物体层遮挡区域的效果。

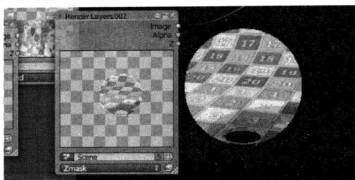


图 7-36 Negate Zmask 反向遮罩的渲染效果

### 7.3.2 通道输出

每一个渲染图层都包含了多条 Pipeline Passes(通道管线) 输出, 每条通道都提供了特定的属性参数, 例如色彩和阴影信息。系统的默认渲染结果是自动叠加全部的通道属性, 也可以将通道分离出来, 在结点系统中分别调整各属性的细节。如图 7-37 所示为 Passes(通道) 的参数面板, 当激活通道名称旁的箭头选项时, 结点中的渲染输入层结点上就会出现对应的通道输出结点。其中部分结点是无法直接查看输出效果的, 需要使用一些附加结点来显示其输出值。

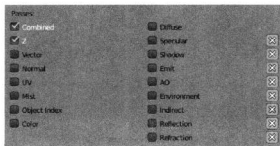


图 7-37 Passes (通道) 参数面板

**Z:** 输出场景的 Z 缓冲通道, 它直观地标识了从摄像机到场景的纵向深度, 可以用来控制画面的景深。如图 7-38 所示为场景中的 Z 通道输出效果。



图 7-38 Z 通道的输出效果

**Vector:** 速度通道, 用于记录物体的运动数据, 需要配合运动模糊结点来调节物体的运动模糊效果。如图 7-39 所示为一个标准运动模糊的结点 workflow。

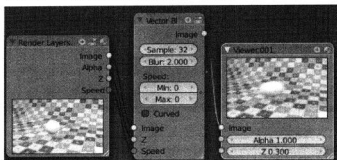


图 7-39 Vector (速度) 通道的输出效果

**Normal:** 法向通道, 常用于输出场景法向通道的效果, 用于后期的二次光照合成。如图 7-40 所示为配合结点调节输出法向图的效果, 其中 Mix 和 Add 结点的灰度值输入需要设置为 7F7F7F。

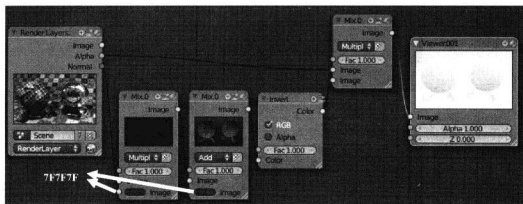


图 7-40 Normal (法向) 通道的输出效果

**UV:** 用于输出当前场景的 UV 坐标, 可以使用结点系统为物体直接映射一个 UV 贴图, 常用于物体的贴图测试渲染。如图 7-41 所示为使用结点系统完成的 UV 替换效果。

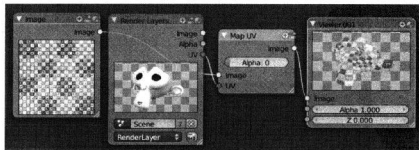


图 7-41 UV 通道输出与重映射的输出效果

**Mist:** 输出雾效的深度值。

**Object Index:** 用于输出特定物体的 Alpha 蒙版, 被选择显示的物体需要提前在 Relations (关系) 面板中指定序列。如图 7-42 所示, 首先为物体设置 Pass Index (通道序列号) 为 1, 那么在输出中可使用结点 ID Mask 来选择指定物体的 Alpha 通道。

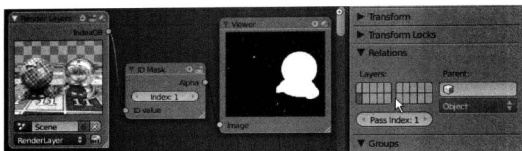


图 7-42 Object Index（对象序列）的输出效果

**Color:** 输出渲染画面的色彩信息，如图 7-43 所示为 Color（色彩）输出的结点预览效果。

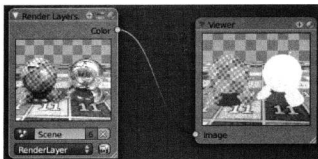


图 7-43 Color（色彩）的输出效果

**Diffuse:** 输出画面的漫反射信息，如图 7-44 所示为 Diffuse（漫反射）通道的输出预览效果。

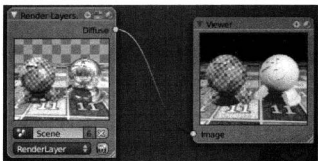


图 7-44 Diffuse（漫反射）的输出效果

**Specular:** 输出画面的高光信息，如图 7-45 所示为 Specular（高光）通道的输出预览效果。

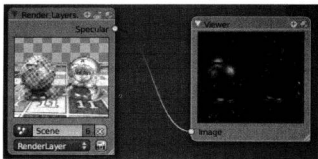


图 7-45 Specular（高光）的输出效果



**Shadow:** 输出场景中的阴影信息, 如图 7-46 所示为 Shadow (阴影) 通道的输出预览效果。

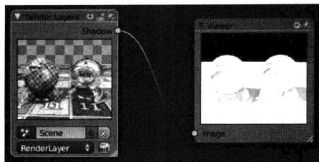


图 7-46 Shadow (阴影) 的输出效果

**Emit:** 用于输出场景中自发光的亮度信息, 如图 7-47 所示为 Emit (自发光) 通道的输出预览效果。

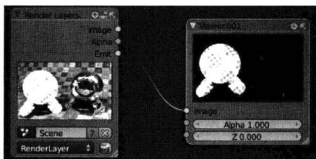


图 7-47 Emit (自发光) 的输出效果

**AO:** 输出 Ambient Occlusion 环境光信息, 这里需要开启 AO 的计算选项才能正常输出。如图 7-48 所示为 AO 通道的输出预览效果。

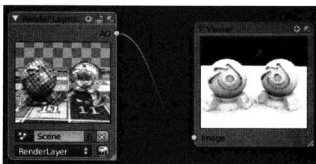


图 7-48 AO 的输出效果

**Environment:** 输出 Environment Lighting (环境光照) 的预览效果。

**Indirect:** 输出 Indirect Lighting (间接光照) 的预览效果。

**Reflection:** 输出物体上的反射光色彩信息, 如图 7-49 所示为 Reflection (反射光) 通道的输出预览。

**Refraction:** 输出物体上的透视色彩信息, 例如为右边球体的基座添加了 Z 向 Transparency (透明) 着色器, 如图 7-50 所示为 Refraction (透视) 通道的输出预览效果。

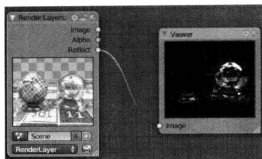


图 7-49 Reflection (反射光) 的输出效果

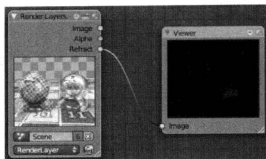


图 7-50 Refraction 透视的输出效果

所有的通道都可以在同一个渲染图层上输出,并在结点系统中执行连接调节,具体的结点调节将在第10章的结点应用中详细讨论。如图7-51所示为一个渲染图层输出的各种通道效果。



图 7-51 场景的多通道输出效果

## 7.4 摄像机

摄像机的属性决定了渲染的画面效果,针对不同的应用需要使用不同类型的摄像机。例如通过模拟鱼眼摄像机来拍摄全景画面,如图7-52所示为Weber State University的Ott Planetarium工作室利用Blender制作的开源球面全景科教电影《Sizing Up Space》。



图 7-52 开源全景电影《Sizing Up Space》

### 7.4.1 显示面板

在视图中单击 RMB 选择摄像机物体，工具面板中会弹出摄像机的相关属性设置。如图 7-53 所示为 Camera（摄像机）的 Display（显示）参数面板。

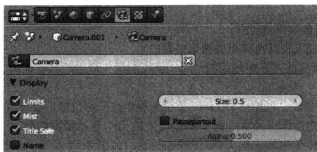


图 7-53 Display（显示）参数面板

**Limits:** 显示摄像机的可视范围和焦距位置，如图 7-54 所示，其中黄线标示了镜头的可视范围，而十字星则标示了焦距的位置。

**Mist:** 开启迷雾效果的显示。

**Title Safe:** 显示安全边框范围，超出边框范围的画面可能在电视屏幕中无法显示出来。

**Name:** 在镜头画面中显示摄像机物体的名称。

**Size:** 控制摄像机在视图中的尺寸。

**Passpartout:** 加深镜头外画面的显示透明度。

### 7.4.2 镜头面板

如图 7-55 所示，为摄像机 Len（镜头）的参数面板。

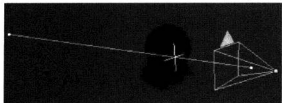


图 7-54 Limits 摄像机的可视范围显示

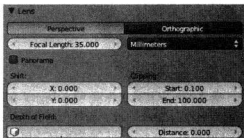


图 7-55 Len（镜头）的参数面板

**Perspective/Orthographic:** 镜头类型，Perspective 为透视图视角，使用 Angle（角度）参数来调整镜头的焦距。如图 7-56 左图所示为 Perspective（透视图）的视角效果。Orthographic 为正视图视角，如图 7-56 右图所示，为同一个镜头在场景中的正视图视角效果，可以使用 Scale（缩

放) 参数来调整场景中物体到镜头的距离。

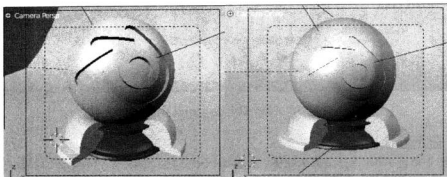


图 7-56 Camera Type (镜头类型)

**Panorama:** 开启鱼眼视角效果。

**Shift:** 在视觉平面上变换摄像机的视角位置。

**Clipping:** 控制镜头的可视距离范围, Start 值控制镜头最近的可视距离, 而 End 值则控制镜头最远的可视距离。

**Depth of Field:** 景深, 可使用 Distance 数值来控制焦距的位置, 也可以使用 Empty (空) 物体的坐标位置数据来定位景深距离。如果要在渲染输出中显示景深的效果, 还需要在结点中添加一些后期处理, 具体使用将在第 10 章中详细讨论。如图 7-57 所示, 左图为无景深设置的效果, 而右图为了使用了景深的渲染效果。

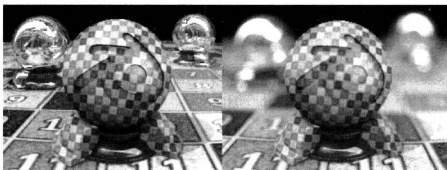


图 7-57 Depth of Field (景深) 效果

## 7.5 网络渲染

### 7.5.1 渲染农场

著名的电影特效工作室 Weta Digital 在制作电影《2012》的过程中, 花费在单帧画面上的渲染时间平均为 20 个小时, 而整个地震场景的特效帧数超过 7000 帧, 所以总共的渲染时间大约为 141 120 小时。与此同时, 电影中还使用了大量的水与火等物理模拟, 光这些特效场景的运算数据量就高达 1.2 PB (即 1200 TB)。这意味着, 如果所有的渲染处理和模拟计算只由一台计算机来完成的话, 大约需要 16 年的时间! 因此, 在工业级的渲染应用中, 常使用由多台计算机并联网组成的网络渲染模式来完成大数据量的渲染和计算输出。电影《2012》的地震特效场面如图 7-58 所示。



图 7-58 电影《2012》的地震特效场面

Cluster（集群）系统最早是由 Pixar 提出的一种名叫“阿尔弗雷德”的多任务阶梯式架构，它将多台计算机通过网络通信协议连接成一个计算机群，把工作负载从一个超载的计算机迁移到集群中的其他计算机上，这一特性称为 Load Balancing（负载均衡）。如图 7-59 所示为一个渲染集群的结构图，服务器可以将渲染任务分成若干个小单元 Job（任务），并通过网络发送给集群上的每一个 Node（结点）计算机执行计算。每当有结点上的 CPU 完成任务并置为空闲时，系统就会向其再分配一个新的任务单元。这种大量的计算机集群是目前被广泛应用于电影业和动画业的渲染模式，我们又亲切地将这种集群简称为 Render Farm（渲染农场）。

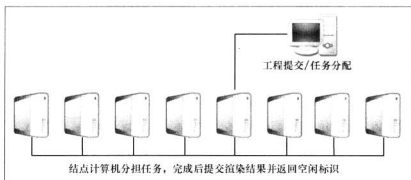


图 7-59 集群系统的结构图

渲染农场不仅需要使用大量的 CPU，还需要使用一套足够用来交换并存储数据的局域网。在电影《阿凡达》中，每一帧的图像大约需要 12 MB 的存储空间，按照 1 秒钟 24 帧计算，每分钟的存储数据就高达 17.28 GB，因此整部电影大约有 3 PB 的数据量。为此 Weta Digital 对传统的渲染农场重新进行了设计，新的系统由 4 个超级计算集群组成，使用了 1280 台 HP 刀片服务器，总内存大约为 104 TB，内联采用的是万兆以太网，数据传输采用了光纤通道，因此整个农场的累积峰值计算性能达到了 205 万亿次浮点运算/秒。由于使用了强大的渲染农场，Weta Digital 不仅能以更快的速度制作最先进的动画产品，而且还能有效地降低运营的成本。电影《阿凡达》的剧照如图 7-60 所示。

总结渲染农场的系统特点如下：它需要十分稳定的网络传输和数据交换，足够的系统存储空间，易于管理的结点任务分配制度，自动报错功能以及异常管理。要保证一套具备商业集群渲染能力的农场并不容易，这不仅需要大量的资金投入，还需要高端的技术支持。但是对于拥有小型局域网的工作室应用，完全可以使用由 Blender 提供的一套十分强大且足够胜任的农场解决方

案,这样就可以十分方便地搭建一套小型网络渲染农场,而无需去购置第三方软件来配置及管理这个平台。



图 7-60 电影《阿凡达》剧照

## 7.5.2 服务器端

要使用网络渲染,应首先在窗口的标题栏上将渲染器选择为 Network Render (网络渲染) 模式,如图 7-61 所示。

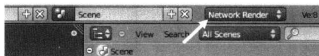


图 7-61 选择 Network Render (网络渲染器)

Master (服务器) 模式用于提供网络渲染的主机服务,它将主持 Job (任务) 的分配和 Node (结点) 间的工程调度,因此服务器端并不会调用当前的主机执行渲染运算。转入工具面板中的渲染选项,如图 7-62 所示为 Master (服务器) 的参数面板。

**Start Service:** 运行服务器,当前的 Blender 将自动切换至服务器模式,成为网络中的主机。

**Path:** 用于选择服务器保存渲染结果、日志文件和提交任务文件的目录。

**Server Address/Port:** 设置服务器的 IP 地址和调用的网络端口,建议这里进行手动设置。如图 7-63 所示为 Master (服务器) 的 Setting (配置) 参数面板。

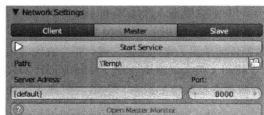


图 7-62 Master (服务器) 端的参数面板



图 7-63 Master (服务器) 的 Setting (配置) 参数面板

**Broadcast:** 开启服务器的广播功能,运行后系统将向本地网络发送主机 IP 和通信端口等信息,使 Slave (负载) 端和 Client (工作站) 端在接收到主机的信息后可以进行自动连接。

**Clear on exit:** 服务器停止后,自动清理本地目录中的所有文件,不建议勾选。

### 7.5.3 负载端

Slave（负载）端主要提供运算支持，它将接受并渲染由服务器发送过来的 Job（任务），完成后再上传给服务器，并返回空闲的指令，等待下一个新任务的分配。如图 7-64 所示为网络渲染中的 Slave（负载）端的参数面板。

和 Master 端的面板选项类似，可以在 Path 目录选项中选择当前主机存放渲染任务和日志等文件的目录。在开启服务前，应首先单击下面的 Refresh（刷新）按钮，搜索网络中的主机广播，一旦完成了捕获，系统将自动把服务器的 IP 和网络端口显示在 Server Address 和 Port 选项中。当单击 Start Service（开始服务）后，Blender 将把当前的主机转换成一个 Node（结点）计算机，并与服务器建立连接，然后等待主机的任务分配。

如图 7-65 所示为 Slave（负载）端的 Setting（配置）参数面板。



图 7-64 Slave（负载）端的参数面板

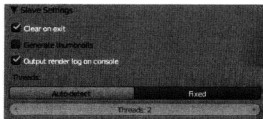


图 7-65 Slave（负载）端的 Setting（配置）参数面板

**Generate thumbnails:** 是否让负载端在渲染时显示渲染的预览画面。

**Output render log on console:** 在负载端的控制台上开启渲染的日志显示。

**Threads:** 设置调用的 CPU 线程数，默认设置为 Auto（自动）模式。

### 7.5.4 工作站端

Client（工作站）端是渲染网络中用于执行任务提交并取回渲染结果的平台，也是我们实际工作和监控整个渲染的系统。如图 7-66 所示为网络渲染中的 Client（工作站）端的参数面板。

Client（工作站）端的配置和 Slave（负载）端类似，首先单击 Refresh（刷新）按钮，用于接收网络中的主机广播，在完成 Path（路径）选择后，即可准备执行任务提交。如图 7-67 所示为 Client（工作站）端的 Setting（配置）参数面板。



图 7-66 Client（工作站）端的参数面板



图 7-67 Client（工作站）端的 Setting（配置）参数面板

**Name:** 提交的任务名称，默认采用原工程文件的命名。

**Category:** 任务的目录，可选选项，用于服务器端的分类管理。

**Priority:** 设置任务的优先级，服务器将根据任务的优先级来执行运算的先后顺序。

**Chunks:** 设置当前文件将被分割为多少个结点计算任务。

### 7.5.5 搭建实例

接下来，我们就来搭建一个小型的渲染农场。首先应保证网络中至少拥有 2 台以上的计算

机,并将其中一台设置为 Master (服务器主机),另外一台设置为 Slave 负载端,接着设置任意一台为一个 Client (工作站端)。由于 Master (主机)并不会执行渲染运算,所以为了能尽可能地利用网络资源,可以采用如下结构做本地实验,如图 7-68 所示。但是这种连接方式仅适合小型的布局,如果网络结点足够多,还是建议架设一台单独的机器作为服务器端。

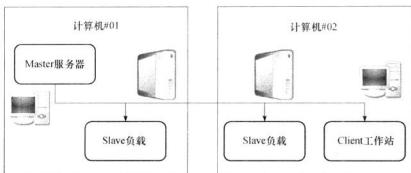


图 7-68 由 2 台计算机组成的渲染农场

接着在#01 主机上启动 Blender 程序,切换模式为 Master (主机模式),并设置服务器目录以及#01 主机的网络 IP 和通信端口,如图 7-69 左图所示。接着单击 Start Service (开启服务),控制台中将显示当前主机正在向局域网广播通信信息,如图 7-69 右图所示。

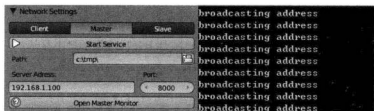


图 7-69 开启服务器端的网络广播

接着在#01 主机上再启动一个新的 Blender 程序,将渲染模式切换为 Slave (负载)模式,单击 Refresh (刷新)按钮,系统将自动捕获网络中的服务器广播,并将服务器的网络 IP 和通信端口显示在窗口中,如图 7-70 左图所示。接着单击 Start Service (开启服务),Server (服务器)的控制台窗口中将显示结点与主机的接入情况,如图 7-70 右图所示。



图 7-70 建立 Slave (负载) 结点

转至#02 计算机上,开启一个 Blender 程序,将其配置成 Slave (负载)结点,并与服务器建立连接。接着启动一个新的 Blender 程序,设置其渲染模式为 Client (工作站)模式,单击 Refresh (刷新)按钮获取服务器的地址,如图 7-71 左图所示。接着可以看到 Job Setting (任务选项)面板中出现了任务提交的界面,如图 7-71 右图所示。



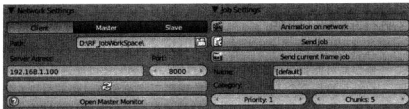


图 7-71 建立 Client (工作站) 结点

**Animation on network:** 发送一个动画工程至网络渲染服务器。

**Send job:** 发送当前工程。

**Send current frame job:** 发送当前工程的帧任务至网络渲染服务器。

单击对应的渲染选项,即可将当前的工程从 Client (工作站) 端发送至 Server (服务器) 端,由服务器将工程拆分,再分发给网络中的各结点 Slave (负载) 端计算机进行渲染。单击 Open Master Monitor (开启服务器监控),使用浏览器打开服务器的 IP 地址即可实时监测渲染情况以及工程的总进度,界面如图 7-72 所示。

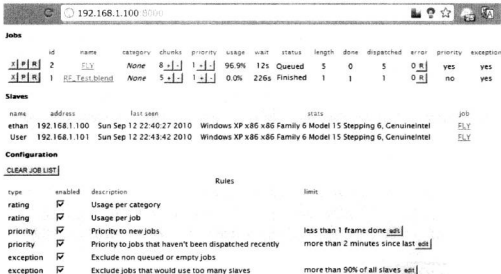


图 7-72 Master Monitor (服务器的渲染监控) 页面

待渲染完成后,即可进入预先设置的文件目录查看渲染成品。

Blender 的内置集群架设相对比较简单,但是足够胜任一个小型工作室的渲染数据量,例如 Ton 已经成功地使用内置集群完成了电影《Sintel》的全部渲染。但是对于更大型项目的网络渲染,还是建议选择更专业的渲染农场软件。目前 Blender 的开发团队正在努力强化该平台的多项功能,例如优化对网络错误的监控与纠错,对物理模拟的数据处理,以及改进网络通信协议等。

如果搭建的是一个纯粹用来服务于渲染的计算机集群,也许并不需要为每台计算机都安装显示器和键盘等输入输出设备,只需要搭载基本的 Linux 系统,并通过远程登录命令行操作,即可完成对服务器或负载结点的建立,当然前提是必须要预先复制一个已经配置好网络等属性的 Blend 工程文件到结点系统中。命令行的参数如下:

```
> Blender -b master.blend -a
> Blender -b slave.blend -a
```

## 7.6 如何提升渲染的性能

渲染是一件令人激动的事情，同时也是一场十分漫长的等待过程。为了获得高质量而且逼真的渲染效果，一幅作品可能会花费数小时的渲染时间，这期间还可能进行反复的测试和调整。对于普通的用户，也许很难拥有令人羡慕的高性能图形工作站和整机房的渲染农场来提高制作效率，但是我们可以适当地对工程做一些优化，软性地提高渲染时的计算性能，加快渲染的速度，减少重复工作和不必要的等待时间。

### 7.6.1 工作环境

提升硬件永远是最简单而且最直接的方式！

Blender 支持最大 8 GB 的系统内存，同时经测试，如果使用 800 MHz 频率的内存，会比使用 667 MHz 的内存提升 30% 左右的性能。所以，如果条件允许，请首先升级内存系统。

Blender 还支持最多 64 个 Threads（线程）计算，所以也尽量升级处理器为多线程的 CPU。

与此同时，请尝试升级图形显卡，虽然这不能提高渲染性能，但是能提高实时 GLSL 预览、雕刻预览以及游戏引擎的渲染效率。

除了硬件的改进，还应该选择配置一个高性能的操作系统。

记住你的工作是制作高品质的 CG，而不是家庭的游戏机或者影音娱乐中心，所以尽可能地使用低资源消耗的操作系統，将尽量多的资源分配给 Blender 使用。如果安装的操作系统是 Windows 系列，请关闭桌面特效以及一些不必要的后台程序。如果可以选择不选择 Windows 系统，那就请选择 Linux 系统。开源的 Linux 系统由于占用资源少、系统稳定、可集成度高以及极其低廉的成本，已经逐渐成为国际上通用而且成熟的电影特效平台，例如，Digital Domain 就是好莱坞最早将工作平台移植到 Linux 系统上的特效公司之一。如果资金充足，还可以考虑 Mac OS 系统，因为苹果系统特别为开发人员设计，良好的兼容性和可靠性能满足艺术家的大部分需要，同时苹果的色彩管理是最符合工业标准的。

当然还可以考虑升级使用 64 位的操作系统，并下载使用 64 位的 Blender，经测试发现，64 位的 Blender 相比于 32 位的版本有 10% 左右的性能优化。

除此之外，在开启渲染时应尽量保持一个干净的计算环境，减少其他并行运行的应用软件，例如不要同时观看电影或者打开浏览器，尽量把系统的资源和 CPU 时间都分配给渲染器。

### 7.6.2 工程配置

优化渲染环境的同时，还可以优化调节 Blender 的渲染配置。

在系统配置中适当地增大系统和 OpenGL 的内存使用限制，这可以在系统参数面板中设置。

针对硬件使用环境，尝试下载源码并自行编译 Blender，通常情况下这可能会使性能提高 20% ~ 40%。如果不熟悉本地编译，也可以去下载针对不同硬件环境而优化的特殊编译版本。

对于 Blender 工程文件，调整一些对象的使用参数也能在一定程度上加快渲染的速度。

删除不需要的灯光物体，或者将它们移到隐藏的层中，因为多余的光照会产生不必要的计算。同时关闭 Lamp（灯光）的阴影选项，由于它的发散特性，产生的阴影会增加渲染的计算量，建议使用 Sun 或 Spot 等方向性更强的灯光来制作阴影。尽量少使用 Area（面）光源，并降低其 Sample（采样值），缩短灯光的照明范围。

对物体的纹理应合理地使用 Baking（烘焙）来做预计算处理，这可以减轻物体的材质计算量。如果没有必要，请取消使用 SSS 等需要预计算的材质。根据需求考虑使用较低的贴图尺寸，例如一个 256 × 256 的贴图只会花费 2048 × 2048 贴图 10% 的内存使用量。

进入物体的编辑模式，移除物体多边形中的重合点。在修改器面板中，降低细分修改器的渲

染细分等级。如果制作的是静帧物体渲染，可以考虑删除一些看不见的物体多边形结构。

可以尝试将摄像机调节为正视图视角，然后使用 ShiftX/Y 来手动切割渲染画面，最后导入到 Photoshop 等软件做拼接合成。同时，降低摄像机的最远可视距离。

在选择渲染输入层时，仅选择有渲染物体所在的层，无用的层尽量不要输入到渲染流程中，以减少数据的预存储。同时尽量使用分层渲染，使用结点做后期的合成管理。

设置渲染属性时，可以关闭 AO 等环境渲染计算，关闭 Edge（边）渲染等后期处理。关闭无用的渲染输出通道，关闭光线追踪的计算。尝试降低抗锯齿输出，或者关闭掉。同时使用较大数值设置的 Tiles 参数来切割画面，减小 CPU 渲染的最小单元尺寸。最后考虑关闭 Alpha 通道的渲染输出，因为这会比只渲染 RGB 色彩消耗更多的内存。

在使用结点制作后期合成时，如果是静止的画面，那么背景层也许只需要一次性的渲染，而不需要重复计算输出，这时可以在结点中使用一个图片结点单独将其导入。

尽可能地分离出高光和阴影等光线效果通道，使用结点来分别调整和强化，避免过分依赖着色器和灯光搭配的默认效果。同时尝试将不同位置的灯光效果分开渲染，使用结点来叠加合成。

最后，希望你能够在这些建议中找到合理配置最优化渲染的方案。当然要想获得最好的渲染效果，还是需要付出一定时间的代价。例如，如图 7-73 所示为 Massimo Righi 花了超过 10 个小时渲染的作品 Green Frog。



图 7-73 Massimo Righi——Green Frog





## Part2 第二部分

# 高级篇

与传统的动画不同，CG 动画是借助计算机来制作的二维或三维数字动画。作为一款优秀的动画套件，Blender 提供了十分高效的动画生产工具集，同时 Blender 还集成了十分强大的物理特效模拟和游戏开发环境。在接下来的 5 章中，将分别介绍动画的基本原理和相关工具、粒子和流体等物理模拟、后期的结点合成工具和序列编辑器、Python（脚本）基础和 Blender 的扩展应用以及游戏引擎的初级入门。

- 第 8 章 动画工具
- 第 9 章 物理特效与仿真
- 第 10 章 后期合成
- 第 11 章 Python 脚本扩展
- 第 12 章 游戏引擎



## 动画工具

动画是一种依靠人眼的视觉残留效应，将静止画面转变为动态画面的艺术。传统动画是将每一帧的动作都在纸上绘制出来，然后使用特殊的拍摄设备将每一幅画面都转换为胶片，再按照每秒24帧的速率在屏幕上进行播放。如图8-1所示，为一个传统动画中的运动分解图，关键帧动画师会将图中动作的关键帧绘制出来，再交给过程帧动画师，来绘制过渡部分的画面。这样的生产流程十分耗费人力和时间，制作一部大型动画片常常需要花费上百名动画师数月的时间，才能完成全部画面的绘制。

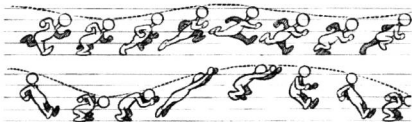


图8-1 动画的运动分解图

3D动画的制作流程与之类似，但是动画师只需要在场景中制作关键帧的动作，剩下的中间帧将交由系统来自动计算生成，这种自动化流程大大降低了人力的成本，同时也节约了生产制作所需要花费的时间。如图8-2所示，为电影《Big Buck Bunny》的关键帧画面。除此之外，CG动画的另外一个优势就是它能创造出更多的特效画面，制作出传统手绘无法完成的光影效果。



图8-2 电影《Big Buck Bunny》的关键帧画面

### 8.1 基本动画工具

#### 8.1.1 曲线编辑器

F-Curve 是用于控制对象的属性，使其产生动画效果的一种工具。物体的每个属性都可以被

定义为一条以时间轴为横坐标, 属性值为纵坐标的 F-Curve (样条曲线)。在不同的时间点上更改这一点的纵轴坐标位置, 即可改变当前时刻物体的属性值。通过按照时间顺序的回放, 系统将根据属性值之间的差值来产生出动画效果, 其中时间轴上的这些属性控制点就被称为关键帧。如图 8-3 所示, 为 Blender 中编辑曲线的 F-Curve Editor (曲线编辑器) 界面。

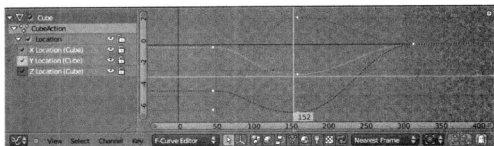



图 8-3 F-Curve Editor (曲线编辑器) 的界面

可以手动添加关键帧, 系统将自动完成过程帧的计算。改变关键帧之间的曲线形态, 就是改变关键帧之间的动画过渡效果, 平滑的曲线过渡为系统的默认设置。在 Blender 中, 所有的对象属性都可使用 F-Curve 来控制, 并制作出动画效果。常用的属性类型包括场景属性、世界属性、结点属性、物体属性、材质属性、灯光属性、纹理属性和相机属性, 可以在状态栏的  面板上, 开启不同属性的显示方式。

## 8.1.2 创建关键帧

Keyframes (关键帧) 有两种类型: 第一种为窗口中的物体形变和位移等属性变化关键帧, 可以直接在场景中执行创建操作; 而第二种为材质或粒子等参数的关键帧, 需要在面板中的选项菜单上执行创建操作。

### 8.1.2.1 在视图中创建关键帧

针对视图中的物体, 可以首先选择物体, 使用常用的形变工具调节形态, 完成后单击快捷键 I, 在弹出菜单中选择需要添加的关键帧类型, 例如 Location (位移)、Rotation (旋转) 和 Scaling (缩放) 等, 如图 8-4 所示。

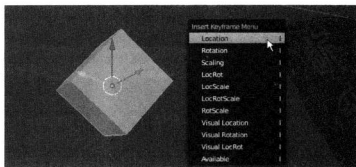


图 8-4 在视图中添加运动关键帧

也可以在编辑器状态栏的 Key 帧菜单中, 使用 Insert Keyframes (插入关键帧) 按钮为当前物体的状态添加关键帧, 如图 8-5 左图所示。Blender 的 Timeline (时间轴) 管理器还提供了自动记录关键帧的功能, 只需要激活状态栏上的红色自动记录按钮, 系统即可根据所选物体的动态情况, 在当前的时间点上自动添加关键帧, 如图 8-5 右图所示。





图 8-5 记录关键帧的两种常用方式

### 8.1.2.2 在参数面板中创建关键帧

而对于第二种类型的关键帧,则需要通过面板选项的参数按钮创建。例如材质的 Diffuse 值按钮,单击 RMB 弹出新建关键帧的菜单,如图 8-6 所示,选择 Insert Keyframe (添加关键帧) 即可。

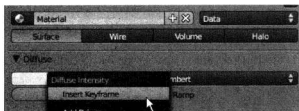


图 8-6 添加关键帧选项

这时转至编辑面板,如图 8-7 所示,可以看到窗口中自动新增了一个 Material (材质) 参数通道,通道中的 Diffuse 曲线即可用于控制物体在一段时间内,材质漫反射值的变化效果。

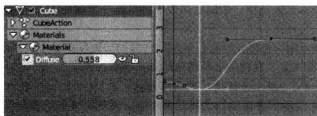


图 8-7 材质的参数动画曲线

### 8.1.3 编辑曲线

使用 RMB 可选择页面中需要编辑的曲线,单击 Tab 即可进入其编辑模式,如图 8-8 所示。进入编辑模式的曲线将在关键帧的点上显示两个操纵杆,用于控制曲线上当前位置点的曲率,可以直接使用 Ctrl + LMB 在曲线上添加新的控制点。曲线上的编辑方式和场景中的网格编辑方式类似,可以使用快捷键 G 来移动当前选择的关键帧控制点,快捷键 R 则用于旋转控制杆,而快捷键 S 则用于缩放当前的控制杆尺寸。

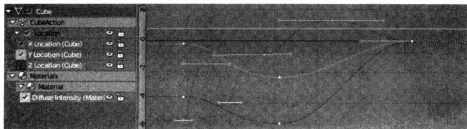


图 8-8 曲线的编辑模式

### 8.1.3.1 Handle (控制杆) 类型

选择控制杆后, 可使用快捷键 H 来修改杆的控制类型。如图 8-9 所示, 单击快捷键 H 后可弹出 Set Keyframe Handle Type (关键帧控制杆) 的类型菜单, 在其中可设置控制杆为 Free (自由) 控制型、Vector (矢量) 控制型和 Aligned (吸附) 控制型等。



图 8-9 设置控制杆的类型

### 8.1.3.2 View Properties (视图的属性) 面板

在当前窗口中单击 N 可调出曲线编辑的 View Properties (视图的属性) 面板, 如图 8-10 所示。



图 8-10 View Properties (视图的属性) 面板

**Show Cursor:** 显示光标选项, 激活后面板中将出现一个绿色的十字光标, 用于显示当前控制点的位置。

**Cursor from Selection:** 可将光标定位至当前选择的关键帧控制点位置上, 而下面的 To Keys 选项可根据自定义的输入值, 在曲线上精确地添加关键帧控制点。

### 8.1.3.3 Active Keyframe (激活关键帧) 的属性面板

在 Active Keyframe (激活关键帧) 的属性面板中, 可对关键点间的曲线 Interpolation (插值) 类型做修改, 这里可选择 Constant (常值) 型、Linear (线性) 型和 Bezier (贝济埃) 型。也可以直接使用快捷键 Shift + T 在窗口中做快捷修改, 如图 8-11 所示。



图 8-11 Active Keyframe (激活关键帧) 的属性面板

**Key:** 帧菜单, 使用数值输入来定位帧的坐标值。

**Handle:** 控制杆菜单, 使用数值输入来调整控制杆的位置。

### 8.1.3.4 曲线修改器

除了手动地编辑曲线, 还可以添加一些 Modifier (修改器), 来快速制作特殊的曲线效果。

如图 8-12 所示,单击快捷键 Shift + Ctrl + M 即可弹出修改器菜单,例如选择 Noise (噪音) 修改器,曲线将被自动添加随机的波噪属性。

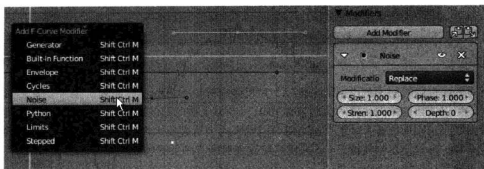


图 8-12 曲线修改器

### 8.1.4 通道管理

物体的每一个属性值都用一个通道来表示,各种通道在 F-Curve 管理器中又以分类的方式来显示,例如 Location (位移) 的属性包括 3 个子坐标数据通道 X、Y 和 Z。单击状态栏上的 Channel (通道) 菜单,可弹出通道的相关管理工具,如图 8-13 所示。

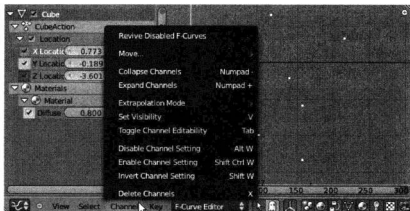


图 8-13 Channel (通道) 管理菜单

**Delete Channels:** 删除当前通道,快捷键为 X。

**Invert Channel Setting:** 翻转对通道的控制操作,例如在编辑模式与曲线模式之间做切换操作,快捷键为 Shift + W。

**Enable/ Disable Channel Setting:** 开启或关闭通道的属性设置,快捷键为 Shift + Ctrl + W 或 Alt + W。

**Toggle Channel Editability:** 进入当前选择通道的编辑模式,快捷键为 Tab。

**Set Visibility:** 设置通道的可视属性,快捷键为 V。

**Expand/Collapse Channels:** 收起或展开当前类型的所有通道,快捷键为 NUM + 和 NUM -。

### 8.1.5 时间轴

#### 8.1.5.1 时间定位与回放

Timeline (时间轴) 面板用于快速查看时间轴上设置的关键帧位置,以及控制回放的时间区

间，它将和曲线面板做同步的回放操作。如图 8-14 所示，为 Timeline（时间轴）的窗口面板。

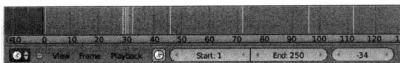


图 8-14 Timeline（时间轴）的窗口面板

时间轴面板以时间为进度值的横坐标，黄色竖线标记了这一时刻时间帧上是否设置有关键帧，绿色竖线标识了当前帧的位置。使用鼠标 LMB 可在轴上做时间的定位，而按住 MMB 可移动时间轴的视图区间，如果配合 Ctrl 组合键可对时间轴区间进行缩放操作。当时间轴的长度超过当前的显示范围时，可以使用快捷键 Home 来快速缩放查看所有的关键帧。

时间轴默认以帧数为计量单位，可以使用快捷键 Ctrl + T 切换至使用秒为计量单位。

时间轴的回放功能包括正常播放和快进等，如图 8-15 所示，单击对应的按钮或使用快捷键 Alt + A，可启动当前场景中的动画回放。



图 8-15 动画的回放面板

### 8.1.5.2 Marker（标记符）

时间轴中最重要的功能之一，就是可添加用于标记时间点的 Marker（标记符）。如图 8-16 所示，为标记了 Marker（标记符）的时间轴窗口面板。



图 8-16 时间轴的 Marker 管理

Marker 常被用于辅助角色动画的制作，例如配合制作镜头的切换、记录场景时间点和配乐配音的制作等。当在时间轴上创建一个 Marker 时，Blender 的其他相关动画模块，例如曲线编辑器、NLA 编辑器等都将自动更新并在对应的位置上添加 Marker。单击快捷键 M 即可在当前时间轴上添加一个新的 Marker，并以帧数字做默认命名。可以使用快捷键 G 来移动选择的 Marker，或者使用快捷键 X 删除当前选择的 Marker。

## 8.2 物体动画

物体动画包含两种控制方式——关键帧动画和路径动画。

### 8.2.1 关键帧动画

首先在场景中心位置添加一个物体，并将时间轴定位在第 1 帧的位置，单击快捷键 I 为物体在当前位置添加 Location（位移）和 Rotation（旋转）关键帧，如图 8-17 左图所示。接着跳跃至第 20 帧位置，使用快捷键 G 将物体移动至场景的另一端，并再次单击快捷键 I 添加 Location（位

移) 和 Rotation (旋转) 关键帧, 如图 8-17 右图所示。

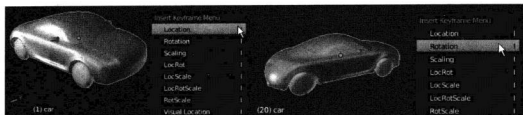


图 8-17 为物体在不同的时间帧位置上添加关键帧

回到曲线编辑视图中, 即可看到系统中已经产生了一条标记为红色的 X 轴 Location (位移) 通道曲线, 曲线上两个关键帧控制点的纵坐标变化, 显示了物体在这段时间内将从 X 轴的负 5 位置移动至正 5 的位置, 如图 8-18 所示。曲线编辑器中的 X 轴代表了动画的播放时间, Y 轴记录了该物体在此坐标方向上的坐标变化值, 由于两点间的默认连线具有曲线特性, 所以物体将产生拥有变速度的移动效果。如果要修改为匀速运动, 可以使用快捷键 H 将曲线类型更改为 Linear 型。

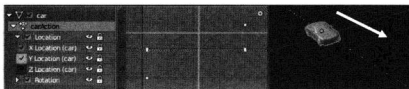


图 8-18 回放物体的位移关键帧动画

接着把时间轴移动至第 10 帧的位置, 此时物体正好运行至坐标原点位置, 将其沿 Y 轴方向移动一定的距离, 并添加一个关键帧, 如图 8-19 所示。这时再单击动画的回放, 可以看到物体产生了一个曲线的运动轨迹动画。

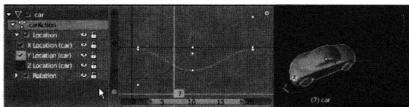


图 8-19 通过增加关键帧来调整运动轨迹

但是, 可以发现这样制作的物体在运动过程中, 车头将始终朝向 X 轴的正方向, 而无法沿着路径方向做转动运动。这是纯关键帧动画中令人头疼的问题之一, 因为它只可以制作一个点到点的过渡动画, 缺少对过程细节的控制。如果要控制车头随路径做跟随的转动, 那么将需要添加更多的关键帧, 如果是复杂的曲线运动路径, 那么需要添加的关键帧数量将十分庞大。解决的方法就是使用路径的方法来辅助完成动画。

## 8.2.2 路径动画

在场景同样的位置处添加一条路径曲线, 单击 Tab 进入其编辑模式, 将形状调整为如图 8-20 所示, 这里要注意路径上的箭头, 它注明了路径的方向和物体的运动方向。

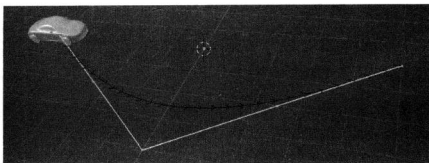


图 8-20 添加 Path (路径)

接着单击物体，并转至其约束面板中，添加一个 Follow Path（路径跟随）修改器，将路径对象的名称输入至目标选项中。这时小车将被自动吸附至路径的起点位置，如图 8-21 所示。

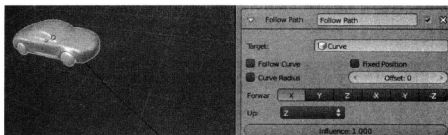


图 8-21 添加 Follow Path（路径跟随）修改器

回放动画，可以发现小车将沿着路径做匀速运动，但是如图 8-22 所示，小车并没有出现车头跟随路径转动的效果。

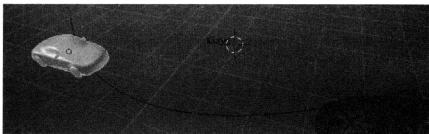


图 8-22 小车跟随路径开始运动

再次选择小车，切换至其修改器菜单，激活 Follow（跟随）选项。这时单击动画回放，就可以看到小车将沿着路径做匀速的跟随运动了，如图 8-23 所示。

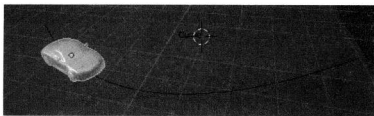


图 8-23 小车沿路径运动

可以直接在曲线的选项面板中控制动画的作用区间，只需要修改 Frames 属性即可完成设置，如图 8-24 左图所示。这里可以将小车完成全部路径的运动时间设置为 20 帧。同时，修改曲线的 Evaluation Time（控制时间）选项即可控制物体运动的速率。首先在其按钮上单击 RMB 添加一个关键帧，接着转至曲线编辑器，调节曲线的形态，这样系统将根据曲线的斜率产生一个自定义的变速度运动效果，如图 8-24 右图所示。

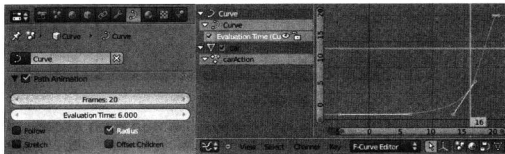


图 8-24 使用曲线来控制物体的运动时间和速度

## 8.3 形变动画

控制一个物体的移动和旋转是最基本的运动动画，Blender 的另一个高级动画功能就是可以制作物体的形变动画。

### 8.3.1 形变关键帧

Shape Keys（形变关键帧）用于存储物体的外形状态，是为物体分别设置不同形状的一类关键帧。物体通过状态之间的切换即可产生形变效果动画，例如常使用 Shape Keys 来制作角色的面部表情动画。

如图 8-25 左图所示，首先在场景中添加一个默认的 Cube 物体，然后切换至物体的 Shape Keys（形变关键帧）面板，如图 8-25 右图所示。单击旁边的加号新建一个初始关键帧，默认命名为 Basis。这个关键帧用于存储物体形变的初始状态，任何形变都将以 Basis 关键帧的网格为变形基础。

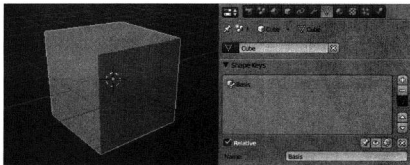


图 8-25 创建 Basis（初始）关键帧

接着再单击加号建立新的形变关键帧，命名为 Key1，它将复制 Basis 的网格形态，如图 8-26 左图所示。单击 Tab 进入当前形状的编辑模式，移动网格上的任意一个点，可以对物体做一个简单的变形操作，效果如图 8-26 右图所示。

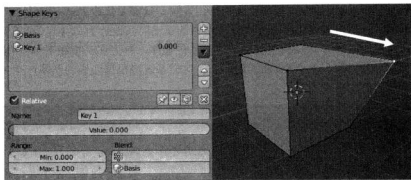


图 8-26 编辑 Key1 的物体形状

这时如果单击 Tab 退回至物体模式，就会发现在 Key1 下做的形状修改将会消失掉，物体被重置为 Basis 关键帧时的形状。这是因为没有设置变形的过渡比例 Value 值，如图 8-27 左图所示。如果要启用 Key1 对物体形状的影响，可以在面板中调节 Value 值，当值不断增大时，物体将会逐渐从 Basis 的形状过渡至 Key1 关键帧下的物体形状，如图 8-27 右图所示。

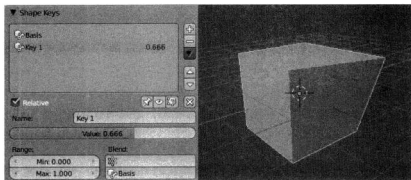


图 8-27 调节 Value 值产生关键帧之间的形变过渡效果

还可以调节面板中的 Range（形变范围）设置，来控制变形的过渡极限，同时 Blend 混合参数还可限制受形变影响的网格为指定的顶点组。

Blender 支持多个关键帧同时作用于一个物体上。如图 8-28 左图所示，再在 Shape Keys 面板中添加一个形变帧 Key 2。单击 Tab 进入编辑模式，然后选择任意一个点对物体做修改操作，效果如图 8-28 右图所示。

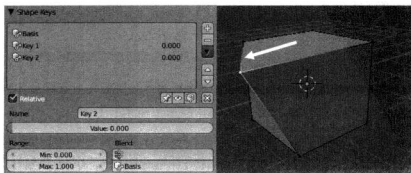


图 8-28 添加第二个关键帧 Key 2



接着同时修改两个关键帧的 Value 值，可以看到物体的形变将产生叠加效果，并且两个不同关键帧之间的变化不会受各自参数设置的影响，效果如图 8-29 所示。

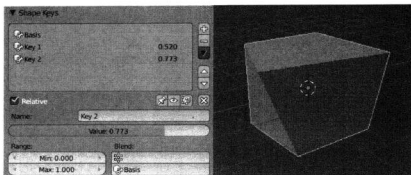


图 8-29 多关键帧在物体形变上的叠加操作效果

也可以使用 Dope Sheet（动作编辑器）来快速编辑关键帧之间的动画过渡，只需要在 Dope Sheet 中切换至 ShapeKey Editor（形变帧编辑器）视图即可。如图 8-30 所示，为当前物体的关键帧界面。



图 8-30 ShapeKey Editor（形变帧编辑器）的界面

将时间轴移动至第 10 帧，然后将两个 Key 关键帧的 Value 值稍作调整，如图 8-31 所示，物体将在窗口中产生同样的形变动画。

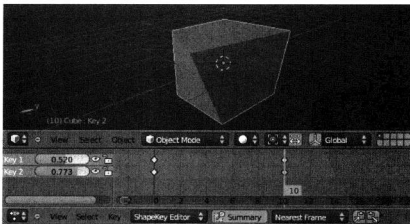


图 8-31 使用 ShapeKey Editor（形变帧编辑器）来管理形变动画的关键帧位置

当然还可以使用 F-Curve Editor（曲线编辑器）来调整属性曲线的斜率，用于控制变形的速率快慢，如图 8-32 所示。当为一个 Key 建立关键帧时，F-Curve Editor 将自动产生一个曲线通道，

用于存储当前的关键帧信息。

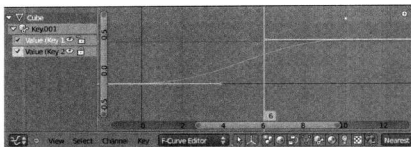


图 8-32 使用 F-Curve Editor 来编辑形变动画的属性曲线

### 8.3.2 形变驱动器

前面介绍了 3 种方法来控制物体的形变，但是功能最强大的还是使用第 4 种 Driver（驱动器）来控制形变，这样可以十分直观地在场景中使用骨骼或其他物体来控制物体的形变动画。

首先，建立一个 Key 3 关键帧，单击 Tab 进入编辑模式后，对物体做一些简单的形变操作，效果如图 8-33 所示。

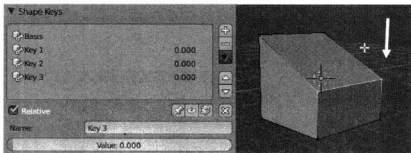


图 8-33 新建关键帧 Key 3

本例中以骨骼为控制物体，当然还可以选择 Empty 或 Mesh 等其他物体。如图 8-34 左图所示，首先在物体的旁边单击 Shift + A 新添加一个骨骼物体，接着按住 Shift + RMB 选择 Cube 物体，在其 Key 3 关键帧上单击 RMB，选择 Add Driver（添加驱动器），如图 8-34 所示。

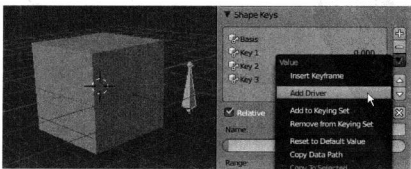


图 8-34 为 Key 3 形变关键帧添加 Driver（驱动器）

接着在 Graph Editor（图形编辑器）中，切换至 Driver（驱动器）视图，如图 8-35 所示。单击 N 弹出驱动器管理的属性面板，将驱动器 Type（类型）设置为 Average Value（平均值）。

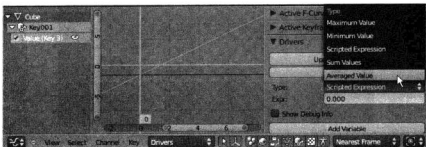


图 8-35 设置驱动器的类型

接着单击 Add Variable (添加驱动变量), 为驱动器指定一个控制对象, 并设置属性为 Transform Channel (形变通道), 如图 8-36 左图所示。由于此时还没有设置控制物体, 所以面板中会显示一个 Error (错误) 信息。接着在 Ob/Bone (对象和骨骼) 面板中, 输入用于控制当前形变的骨架和骨骼物体名称, 并设置骨骼的有效控制方向为 X 轴, 激活 Local Space (本地空间) 选项, 具体参数如图 8-36 右图所示。

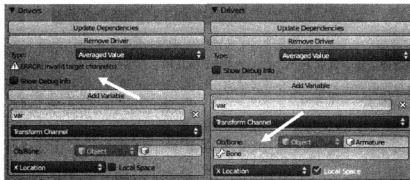


图 8-36 设置驱动器的控制物体

切换回视图窗口中, 这时再选择骨骼物体, 单击 Ctrl + Tab 进入其 Pose (姿势) 模式, 如图 8-37 左图所示。由于设置的控制方向为 X 轴方向, 那么单击 G + X 将骨骼沿 X 坐标做移动操作时, 物体将在骨骼的驱动下自动执行 Key 3 的关键帧形变动画, 变形的幅度由骨骼的位移距离来决定, 如图 8-37 右图所示。

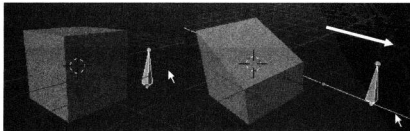


图 8-37 使用骨骼物体来驱动形变动画

### 8.3.3 晶格形变

Lattice (晶格) 形变是使用网格形状的 Lattice (晶格) 物体, 来改变被控对象变形效果的一种动画方法。首先新建一个动画物体, 如图 8-38 所示。

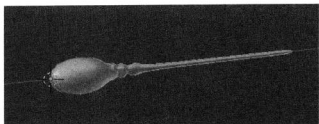


图 8-38 新建一个动画物体

接着单击快捷键 Shift + A 添加一个 Lattice（晶格）物体，如图 8-39 所示。

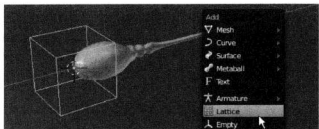


图 8-39 添加 Lattice（晶格）物体

在晶格的属性面板中找到 UVW 选项，分别调整晶格 3 个坐标轴上的细分等级，如图 8-40 左图所示。针对角色的需求，晶格细分等级越高，控制点数量将越多。调整晶格最后的细分等级效果如图 8-40 右图所示。

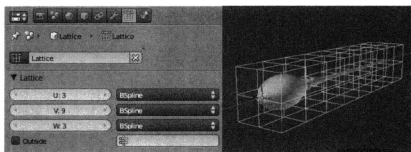


图 8-40 调整 Lattice 晶格的细分属性

接着选择角色物体，并单击 Shift + RMB 同时选择晶格物体，然后使用快捷键 Ctrl + P 建立父子关系，如图 8-41 左图所示，选择封套模式为 Lattice Deform（晶格封套）。这时单击 Tab 再进入

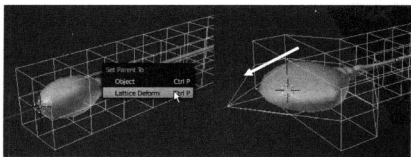


图 8-41 修改晶格结构来控制被控对象的形变

Lattice（晶格）的编辑模式，即可通过调整晶格上的点阵排列，来控制对象角色的形状变化了，效果如图 8-41 右图所示。

### 8.3.4 钩镰形变

由于并不需要使用晶格上全部的控制点，同时每次都要进入晶格的编辑模式，来调整被控对象的外形，显得十分麻烦。这时可以使用 Hook（钩镰）工具，来自定义一些特定的晶格控制点。首先进入晶格的编辑模式，选择将要使用的主要控制点，单击快捷键 Ctrl + H 添加一个 Hook（钩镰），选择模式为 Hook to New Object（建立一个新的 Hook 外部控制对象），如图 8-42 所示。

单击 Tab 退出晶格的编辑模式，这时就可以看到在刚才点的位置上，系统默认添加了一个 Empty 物体作为钩控制器，如图 8-43 所示。这样就只需要在物体模式下控制 Hook 物体，即可实现晶格对对象外形变化的调整操作了，也使得变形动画的操作变得更为方便和快捷。

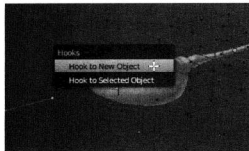


图 8-42 建立 Hook（钩镰）物体

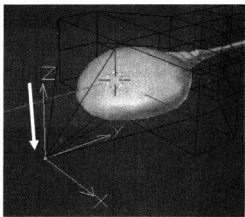


图 8-43 使用 Hook（钩镰）控制器来调整对象的外形变化

## 8.4 骨骼系统

在制作人物等角色模型时，为了塑造虚拟人物的姿态或动作动画，常需要参照现实中的生物体结构，为虚拟人物添加 Armature（骨骼）系统，来支撑角色完成特殊的姿势和动作。如图 8-44 所示，为虚拟人物和用于调整其动作的骨骼结构效果图。

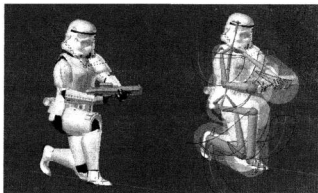


图 8-44 人物姿势与骨骼结构的效果图

骨骼的概念不难理解，它和现实中生物体的骨骼功能类似，用于支撑网格结构，带动躯干做运动或转动。一个 Armature（骨骼）可以由多个 Bone（骨头）组成，每一个生物体的骨骼都不同。如图 8-45 所示，可以在视图窗口中单击 Shift + A 添加一个标准的人体骨骼。

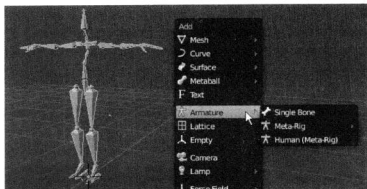


图 8-45 添加 Armature（骨骼）

Bone（骨头）是 Armature（骨骼）的最小组成单元，它具有原心等基本的物体属性，能够实现移动和转动等形变动作。和编辑其他物体的模式不同，骨骼有三种编辑模式，如图 8-46 所示，从左至右分别为骨骼的 Object（物体）模式、Edit（编辑）模式和 Pose（姿势）模式。

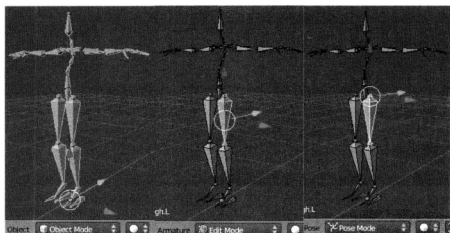


图 8-46 骨骼的三种编辑模式

## 8.4.1 编辑模式

### 8.4.1.1 骨骼基本操作

Object（物体）模式下，骨骼将被定义为一个物体，操作方式和其他类型的物体一样。单击 Tab 可进入其 Edit（编辑）模式，用于制作骨骼的结构，调整骨骼的数量。如图 8-47 所示，为编辑模式中的 Armature（骨骼）菜单。

首先使用快捷键 X 删除全部骨骼，然后单击 Shift + A 在光标位置处建立一根新的骨骼，如图 8-48 左图所示。默认的骨骼是一个八面体物体，较细的一端叫 Head（头），较粗的一端叫 Tail（尾）。骨骼的角度和尺寸都是由头部和尾部的相对位置所决定的，骨骼的中间部分将根据首尾的状态自动连接，因此对骨骼的修改实际上就是分别对骨骼头尾部分的修改。如图 8-48 右图所示，为骨骼的 Transform（形变）菜单。

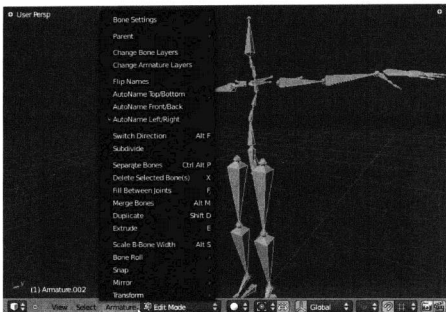


图 8-47 骨骼的 Edit Mode（编辑模式）

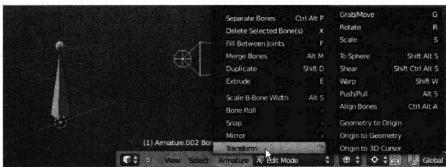


图 8-48 骨骼的形态和 Transform（形变）菜单

单击快捷键 E 可在头和尾的位置上分别挤压出新的骨骼，如图 8-49 左图所示。骨骼的编辑操作同网格物体类似，选取骨骼头和尾后，按 G 使用移动操作即可调节骨骼的尺寸大小，也可使用快捷键 S 对骨骼整体做缩放操作，如图 8-49 中图所示。如果选取骨骼整体并单击 R，可执行以原心为原点的转动操作，如图 8-49 右图所示。

#### 8.4.1.2 骨骼的命名规范

制作骨骼时最重要的一点就是对骨骼做命名操作，这涉及到后面对角色绑定与蒙皮操作时的工作效率。一般来说，角色的骨骼都会以对称的方式进行建模，首先制作一个简单的对称骨架，如图 8-50 所示。当开启所有骨骼的命名显示时会发现，随意建立的骨骼名称十分混乱，如果一个角色有几十根这样类似非规范命名的骨骼，那将给管理和绑定调整带来很大的麻烦。

AutoName（自动命名）功能提供了 3 种模式，可以使用快捷键 W 调出自动命名菜单。其中 Top/Bottom 可将骨骼按照上下结构进行命名，Front/Back 则按照前后结构命名。如图 8-51 所示，使用 Left/Right 对称命名，可以看到对称的骨骼部位都分别在后缀上添加了 L 和 R，用于标记左右结构的区别。

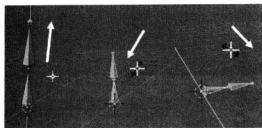


图 8-49 骨骼的基本操作方式

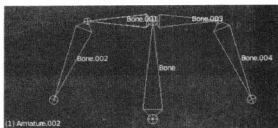


图 8-50 对称的骨骼结构

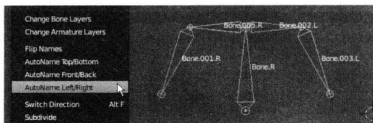


图 8-51 骨骼 AutoName（自动命名）模式

### 8.4.2 姿势模式

在 Pose（姿势）模式下，将无法编辑骨骼的形态，但是可以为骨骼添加 Constraints（约束控制）等关系设计，以及完成姿势的调整和动作的编辑等。如图 8-52 所示，为 Pose 模式的 Pose 菜单。

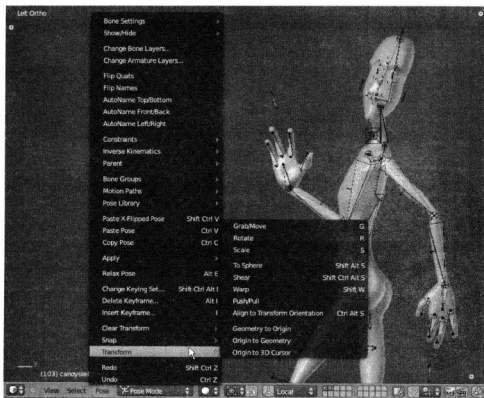


图 8-52 Pose 模式和 Pose 菜单



Pose 模式是制作骨骼动画的环境,因此在选项菜单中还增加了 Insert Keyframe (添加关键帧) 等动画编辑选项,可以使用快捷键 I 为当前骨骼添加关键帧。有关 Pose 模式下的骨骼调节,将在第 14 章中详细讨论。

### 8.4.3 骨骼物体数据选项

在选项面板中,可以找到骨骼的 Object Data (物体数据) 选项,参数面板如图 8-53 所示,可以在这里为骨骼执行命名等操作。

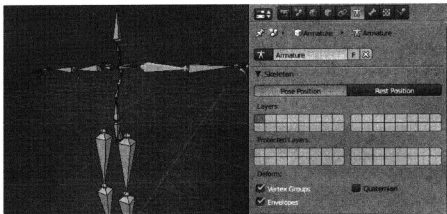


图 8-53 骨骼的 Object Data (物体数据) 选项参数面板

#### 8.4.3.1 Skeleton (骨骼) 属性面板

在骨骼的 Pose (姿势) 编辑模式下, Pose Position (姿势状态) 选项可开启对骨骼的调整编辑,而选择 Rest Position (姿态重置) 功能,则锁定对骨骼任何姿势的调整。骨骼也有 Layers (层) 管理的选项,用于对骨骼做分层存储管理。例如为一个手掌绑定骨骼,可以在同一位置上添加两根骨骼,分别将它们设置为 FK 和 IK 控制模式,并放置放在不同的层中,利于分类管理。

Deform 定义了骨骼与网格之间的关系模式,其中最常用的骨骼与网格绑定工具为 Vertex Groups (顶点组),它使得骨骼可以控制与其同名的顶点组网格。

如图 8-54 左图所示,首先新建一个矩形物体,对其做 4 次细分。接着添加一个骨骼,并挤压出一个新的骨骼 Bone.001。单击 RMB 选择 Cube 物体,再按住 Shift + RMB 选择骨骼,单击 Ctrl + P 调出父子化菜单,这里有几种网格与骨骼建立关联的方式,如图 8-54 右图所示。

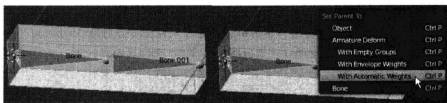


图 8-54 将物体与骨骼建立父子关系

**Object:** 物体选项,仅将物体与骨骼建立对象级别的关联关系。

**With Empty Groups:** 空白组选项,可以为网格自动添加以骨骼名称命名的顶点组列表。

**With Envelope Weights:** 封套权重选项,可以根据 Envelope (封套) 属性,对建立的顶点组自动绘制权重。

**With Automatic Weights:** 自动权重,可以自动绘制骨骼的权重。

**Bone:** 骨骼选项, 可将物体整体都绑定到单个骨骼上。

如图 8-55 所示, 为选择 With Automatic Weights 模式时, 系统自动建立的顶点组和权重图效果。

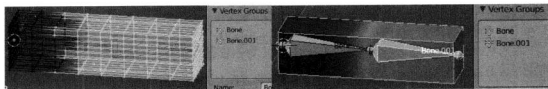


图 8-55 自动生成的顶点组和权重图

顶点组标识了对应骨骼所控制的网格范围, 而权重值则用于标识骨骼形变将带动网格运动的强度。如图 8-56 所示, 在默认的权重模式下, 选择骨骼并进入其 Pose (姿势) 模式, 分别单击快捷键 R 对其做 X 轴和 Y 轴上的转动操作, 可以看到网格上使用同样命名的顶点组区域也同步产生了对应的形变。

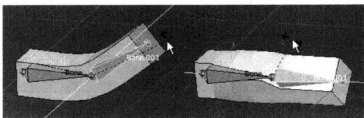


图 8-56 骨骼带动网格做对应的运动

**Envelopes:** 激活封套对网格的影响控制。

**Quaternion:** 启用骨骼的四元数计算, 请参考第 4 章对骨骼修改器的功能描述。

#### 8.4.3.2 Display (显示) 选项面板

如图 8-57 所示, 为骨骼 Display (显示) 选项的参数面板。

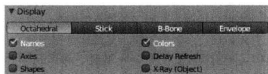


图 8-57 Display (显示) 选项的参数面板

这里定义了骨骼的 4 种显示方式, 如图 8-58 所示, 从左至右分别为 Octahedral (八面体形)、Stick (棍形)、B-Bone (Bezier-Bones) (贝济埃) 形和 Envelope (封套) 形。

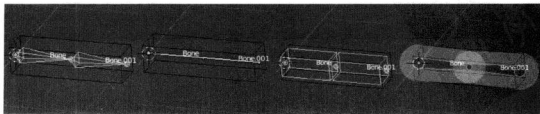


图 8-58 骨骼的 4 种显示方式

**Names:** 在骨骼上名称显示。

**Axes:** 在每个骨骼的端点位置上显示其坐标轴。

**Shapes:** 自定义的骨骼形状显示。

**Colors:** 骨骼组的色彩区分显示。

**X-Ray:** 透视效果显示。

#### 8.4.3.3 Bone Group (骨骼组) 管理选项面板

Bone Group (骨骼组) 管理面板提供了骨骼的分组管理选项, 它的群组管理方式和顶点组类似。如图 8-59 所示, 为骨骼添加一个 Bone Group (骨骼组), 选择一种标识色彩, 单击 Assign 即可将选定的骨骼纳入当前 Group (组) 中。

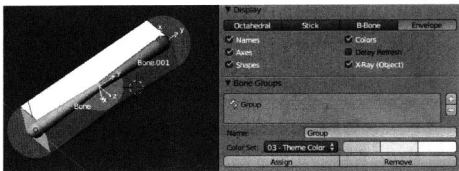


图 8-59 Bone Groups (骨骼组) 的参数面板

#### 8.4.3.4 Ghost (残影) 选项面板

骨骼的 Ghost (残影) 效果可以在场景中显示骨骼的运动轨迹残影, 如图 8-60 所示, 为 Ghost (残影) 选项的参数面板。



图 8-60 Ghost (残影) 选项的参数面板

**Around Frame:** 显示当前帧相邻范围内的运动残影, 其中 Range 用于设置帧的残影显示范围, Step 为显示的帧步长。激活 Selected Only 将只显示当前选择骨骼的运动残影, 取消将显示整体骨骼的运动残影。例如建立一个 12 帧的骨骼转动动画, 如图 8-61 所示, 为当前选择骨骼在相邻 12 帧范围内的运动轨迹残影效果。

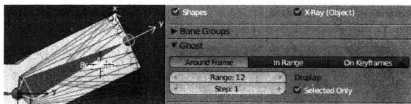


图 8-61 Around Frame (相邻帧) 的轨迹显示效果

**In Range:** 显示指定范围内的运动残影, 其中 Start 设置起始帧的位置, End 设置结束帧的位置, Step 是帧的显示步长。如图 8-62 所示, 为设置一个骨骼在 0 到 12 帧范围内, 画面上产生总共 12 个运动残影的效果图。

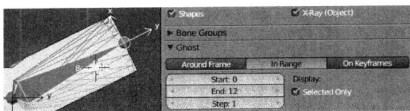


图 8-62 In Range (指定范围) 内的残影显示效果

**On Keyframes:** 仅显示关键帧位置的骨骼残影, 由于这里仅在始末两个位置添加了关键帧, 所以如图 8-63 所示, 为当前关键帧模式下的残影效果图。



图 8-63 On Keyframes (仅关键帧) 的残影显示效果

#### 8.4.3.5 Motion Paths (运动轨迹) 显示选项面板

除了使用残影显示来追踪骨骼的运动关键帧, 还可以使用 Motion Paths (运动轨迹) 显示模式, 来观察骨骼的运动轨迹。如图 8-64 所示, 为 Motion Paths (运动轨迹) 显示选项的参数面板。

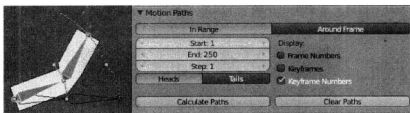


图 8-64 Motion Paths (运动轨迹) 显示选项的参数面板

和残影显示模式唯一不同的地方, 就是要实现轨迹显示需要事先手动 Calculate (计算) 一遍运动路径。如果同时开启 Keyframe Numbers 选项, 关键帧的位置将被高亮显示出来。

### 8.4.4 骨头选项

#### 8.4.4.1 Transform (形变) 选项面板

在编辑模式中, 可以对骨头做外形上的修改和调整, 并制作默认的绑定姿势。如图 8-65 所示

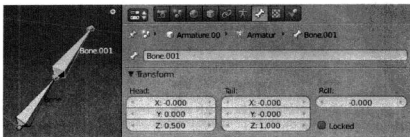


图 8-65 Bone (骨头) 的 Transform (形变) 参数面板

示, 为编辑模式下 Bone (骨头) 的 Transform (形变) 参数面板, 在这里可以直接对选取的每根骨头执行命名和形变等操作。

Head 和 Tail 分别显示了骨头的头尾位置坐标, Roll 则为骨头的本体旋转角度, 单击 Locked (锁定) 选项后, 骨头将被锁定无法执行移动和编辑操作。骨头和网格的编辑方式类似, 选择 Head 后单击 E 可挤压出新的骨头, 如图 8-66 左图所示。但是如果要复制骨头, 则需要选择骨头本体, 单击 Shift + D 来完成新骨头的复制, 如图 8-66 右图所示。



图 8-66 骨头的挤压与复制

#### 8.4.4.2 Relations (关系) 选项面板

新建的骨头将自动与上一级骨头建立父子关系, 如图 8-67 所示, 为骨头的 Relations (关系) 参数面板。

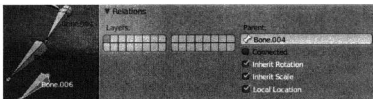


图 8-67 骨头的 Relations (关系) 参数面板

**Layers:** 这里提供了层的查询方式, 用于显示骨头当前被存放的层位置。

**Parent:** 记录了父母骨头的名称, 当关闭 Connected (连接) 选项后, 父子骨头将允许被分离开, 但是视图中会在两者之间显示一条连接的虚线。

**Inherit Rotation/Scale:** 激活后可在建立子骨头的同时, 使其继承父骨头的转动和缩放属性。

**Local Location:** 将骨头定义在本地坐标上。

#### 8.4.4.3 Deform (蒙皮) 选项面板

如图 8-68 所示, 为 Deform (蒙皮) 选项的参数面板, 用于对骨头的封套等特殊选项进行调节操作。

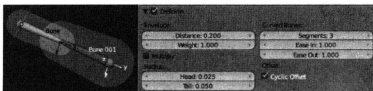


图 8-68 Deform (蒙皮) 选项的参数面板

**Envelope:** 骨头的封套参数, 封套是骨头在物体网格上的影响范围, 只有在封套范围内的网格才会受控于该骨头的形变影响。Distance (距离) 选项用于调节封套的控制范围, 同时 Weight (权重值) 可调节使封套产生形变时的权重值。Distance 值越大, 封套越大, 被控制的网格范围

越大,而 Weight (权重值) 越大,对网格的影响力度也越大。激活 Multiply (重叠) 选项,可叠加封套与顶点组对网格的影响力度。

**Curved Bones:** 这是针对于 B-Bone (类型骨头) 的特殊参数,用于为单个骨头添加分段属性,仅显示在 Pose (姿势) 模式下。在制作脊椎等大曲率和多关节骨架时,没有必要去添加过多数量的骨头来模拟脊椎效果,因为大量的骨头将增加控制的难度。这时就可以使用 Segments (分段) 功能,来将骨头做姿态细分操作,快速而轻松地制作出平滑的多关节骨架。如图 8-69 所示,仅使用两根骨头,分别对其做 3 次分段,最后得到 6 个关节的骨架效果。

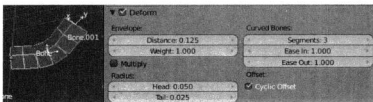


图 8-69 骨头的 Segments (分段) 细分效果

**Radius:** 仅作用于骨头的封套状态,用于控制骨头头和尾的封套半径。

**Offset:** 偏移量,用于控制无父骨头的位置定位。

#### 8.4.4.4 其他参数

当转入到骨头的 Pose (姿势) 模式时,一些参数面板中相应的属性会产生部分调整,例如 Transform (形变) 面板,如图 8-70 所示。

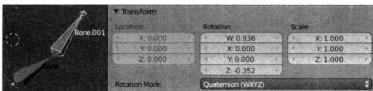


图 8-70 Pose 模式下的 Transform (形变) 面板

可以看到,在 Pose 模式下,Transform 将显示骨头的位置转动和移动参数,而非自身形态的属性。其中 Location 用于显示父骨头的位置坐标,Rotation 和 Scale 分别记录了当前骨头的转动和缩放属性。在属性框上单击 RMB 可弹出参数的关键帧菜单,如图 8-71 所示。

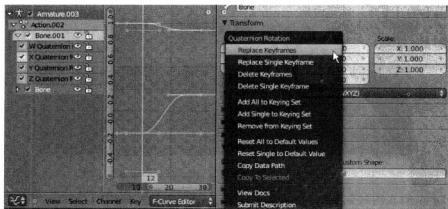


图 8-71 参数的关键帧菜单

如果是没有 F-Curve 控制的属性框将显示为黑白色, 而当存在关键帧的属性框将显示为黄色, 过渡帧的区域则显示为绿色。在这里不仅可以添加 Keyframes (关键帧), 还可以添加 Driver (驱动器) 来控制骨头的形变。

在使用驱动器控制时, Blender 允许建立自定义的控制器形状。如图 8-72 左图所示, 首先建立一个 Circle 物体 Bone\_Cont, 然后选择骨头并单击 Tab 进入其 Pose 模式, 在 Display (显示) 面板的 Custom Shape 中输入自定义的外形名称。再单击 Wireframe 显示框架, 骨头即可被自动替换为刚才新建的 Circle 物体了, 如图 8-72 右图所示。



图 8-72 自定义骨头显示

## 8.5 骨骼动画

常使用 Action Editor (动作编辑器) 和 NLA (非线性) 编辑器来制作骨骼动画, 前者用于管理动画区间中骨骼的运动变化, 而后者则采用非线性编辑的方式将多个动作素材合并在一起, 制作动作块之间的过渡效果。

### 8.5.1 动作编辑器

可以在 Dope Sheet 编辑器中切换至 Action Editor (动作编辑器), 如图 8-73 所示, 为 Action Editor (动作编辑器) 的窗口面板。

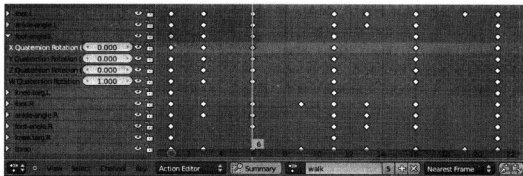


图 8-73 Action Editor (动作编辑器) 的窗口面板

当为一个骨骼新添加一个关键帧时, 系统将自动在 AE 中为其添加一个动作 Channel (通道)。AE 中的通道是对动作帧的宏观管理, 它能标识出当前时间点位置上是否存在一个动作帧, 但是并不会提供动作的具体属性, 这也是其与 F-Curve 曲线编辑器的最大区别。

### 8.5.2 非线性编辑器

当使用 AE 完成了一个动作块的制作后, 就可以使用 Non Linear Animation Editor (非线性动画编辑器) 在一个场景中重复使用该动作块素材, 而不需要再反复地添加片段动画。如图 8-74 所示, 为 Non Linear Animation Editor (非线性动画编辑器) 的面板。

非线性编辑模式同视频的非线性编辑方式类似, 在 NLA 中使用快捷键 Shift + A 即可添加已制作完成的动作片段。其中黄色片段为元素片段, 蓝色片段为两个片段间的动作过渡段, 而紫色片段则为群组后的片段集。

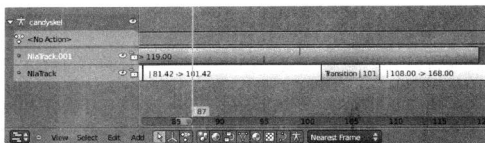


图 8-74 Non Linear Animation Editor（非线性动画编辑器）的面板

## 8.6 约束控制

Constraints（约束）属性定义了物体间的条件约束关系，主要用于为虚拟数字角色添加与环境的互动效果。约束可应用在网格物体上，也可应用于骨骼等特殊物体上，但是并不是每种类型的约束都能应用在骨骼物体上。如图 8-75 所示，为 Constraints（约束）控制器的参数面板。

同物体的修改器使用方法一样，可以为同一个物体添加多个约束控制器，约束的执行顺序是从上向下，可以使用属性控制器上的箭头来调整其在列表中的顺序。如图 8-76 所示，为物体拥有多个约束时的控制器面板。每个约束属性都有一个 Target（目标）对象，用于与当前物体建立约束关系。其中 Influence（影响值）参数控制了约束的强弱效果，值越大，约束的影响力越强。

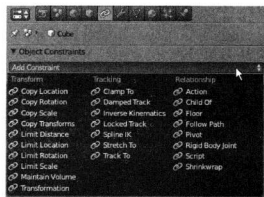


图 8-75 Constraints（约束）控制器的参数面板

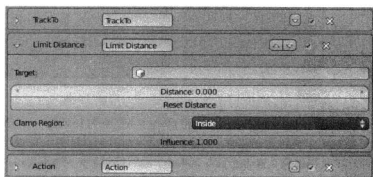


图 8-76 多约束的控制器面板

### 8.6.1 变形约束

#### 8.6.1.1 Copy Location（位移复制）约束

Copy Location（位移复制）约束，可以强迫被约束物体按照目标物体的 Location（位置）数据来定义自身的坐标位置。如图 8-77 所示，添加一个 Empty 作为目标物体，一个球体 Mesh 作为被约束物体，然后选择球体并添加 Copy Location（位移复制）约束。



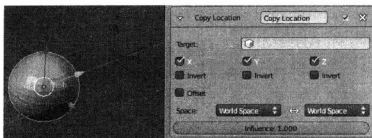


图 8-77 Copy Location (位移复制) 约束的参数面板

**Target:** 用于定义目标物体, 其中 X/Y/Z 分别为目标物体的约束通道, 全选意味着被约束物体将会完全复制目标物体的坐标数据。

**Invert:** 激活后物体将使用反转后的目标坐标数据。

**Offset:** 偏移量, 未激活时被约束物体将直接复制目标物体的坐标数据为自身属性, 两个物体将在坐标上产生重合。当激活 Offset 时, 被约束物体和目标物体之间允许保持一定的偏移量距离, 当目标物体移动时, 被约束物体将把目标物体的位移变化值, 叠加更新至自身的坐标属性上, 产生同步移动的效果。

**Space:** 同步时使用的数据转换标空间, 例如如果要使用目标物体的全局坐标参数, 则应调整参数为 World Space。在一些骨骼应用时, 可能会使用目标物体的本地坐标数据, 那么这时就应该将目标空间修改为 Local (本地) 坐标空间。

如图 8-78 所示, 为 Mesh 和目标物体 Empty 之间添加一定的偏移量, 接着移动 Empty 物体, Mesh 将复制 Empty 的移动幅度到自身坐标上, 做同步的移动。注意, 当被约束物体和目标物体存在偏移量时, 它们之间会显示一条蓝色的虚线来表示约束的关系。

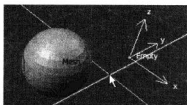


图 8-78 带偏移量的位移复制约束

#### 8.6.1.2 Copy Rotation (转动复制) 约束

和位移复制约束的原理类似, 采用了 Copy Rotation (转动复制) 约束的物体, 将被强制使用目标物体的 Rotation 数据来定义自身的转动参数。如图 8-79 所示, 为在一个立方体上添加的 Copy Rotation (转动复制) 约束。

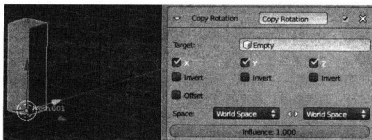


图 8-79 Copy Rotation (转动复制) 约束的参数面板

其中, X/Y/Z 参数将分别控制目标物体的约束坐标轴。转动目标物体 Empty, 如图 8-80 所示, 被约束的立方体也同步产生旋转效果。



图 8-80 转动约束的效果

#### 8.6.1.3 Copy Scale (缩放复制) 约束

Copy Scale (缩放复制) 约束, 可将目标对象的 Scale (缩放) 数据复制到自身参数上。如图 8-81 所示, 为 Copy Scale (缩放复制) 约束的参数面板。



图 8-81 Copy Scale (缩放复制) 约束的参数面板

将目标对象定位在 Empty (空) 物体上, 使用 X/Y/Z 分别控制三个坐标上的缩放比例。其中, 通过设置 Offset (偏移量), 可允许物体在一定的缩放偏移基础上执行约束复制操作。

#### 8.6.1.4 Copy Transforms (形变复制) 约束

Copy Transforms (形变复制) 约束结合了位移复制、旋转复制和缩放复制的数据处理模式, 但是它取消了 Offset (偏移量) 的选项。如图 8-82 所示, 为 Copy Transforms (形变复制) 约束的参数面板。当物体添加了 Copy Transforms (形变复制) 约束后, 将完全使用目标物体的数据来控制自身的所有形变参数。



图 8-82 Copy Transforms (形变复制) 约束的参数面板

#### 8.6.1.5 Limit Distance (距离极限) 约束

Limit Distance (距离极限) 约束可以约束物体与目标物体之间的最大距离值, 如图 8-83 所示, 为 Limit Distance (距离极限) 约束的参数面板。

当与目标物体建立起约束连接后, 它们之间的距离将会被自动计算显示在 Distance 一栏中, 当然, 这个数值也可以手动调整。

**Clamp Region:** 固定方式, 这里总共有 3 种模式, Inside 能将物体之间的距离约束在 Dis-

tance (距离) 范围内, 这里的数值控制了两者之间的最远距离; Outside 则将物体控制在 Distance (距离) 范围外, 这里 Distance 将控制两者之间的最近距离; On Surface 则将两者之间的距离限定在 Distance 数值上, 物体移动将被约束在一个以目标物体为圆心、Distance 值为半径的球体空间表面上。

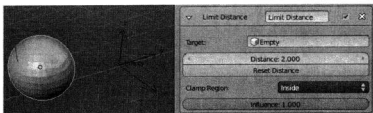


图 8-83 Limit Distance (距离极限) 约束的参数面板

#### 8.6.1.6 Limit Location (位移极限) 约束

Limit Location (位移极限) 约束用于约束物体原心的最大移动范围, 如图 8-84 所示, 为 Limit Location (位移极限) 约束的参数面板。它与 Limit Distance 约束的最大区别在于, Limit Distance 是一个以原心为原点、Distance 为半径的球体约束空间, 而 Limit Location 是以 XYZ 轴为边长的矩形约束空间。

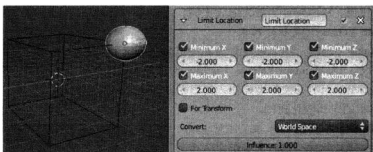


图 8-84 Limit Location (位移极限) 约束的参数面板

原心在每个坐标轴上都可设置一个 Minimum (最小) 位移范围和 Maximum (最大) 位移范围, 这里将它们分别设置为 -2 和 2, 同时在视图中建立一个  $4 \times 4 \times 4$  的立方体。这时再移动球体时, 可以发现它的原心由于被限制了最大移动距离, 将无法移动至这个立方体以外的空间中。

**For Transform:** 使约束可作用于物体的形变效果。

#### 8.6.1.7 Limit Rotation (旋转极限) 约束

Limit Rotation (旋转极限) 约束用于约束物体的最大转动角度。如图 8-85 所示, 为 Limit Rotation (旋转极限) 约束的参数面板。

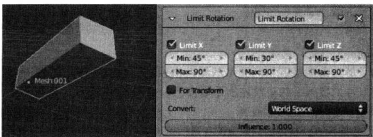


图 8-85 Limit Rotation (旋转极限) 约束的参数面板

和位移极限约束同理，每个旋转坐标轴都有一个角度极限设定值，其中 Min 用于设置最小的转动角度，而 Max 则用于设置最大的可旋转角度。

#### 8.6.1.8 Limit Scaling（缩放极限）约束

Limit Scaling（缩放极限）约束用于约束物体的最大缩放尺寸。如图 8-86 所示，为 Limit Scaling（缩放极限）约束的参数面板。

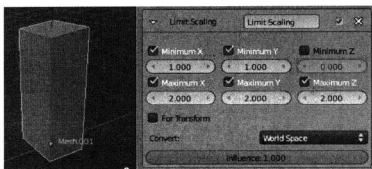


图 8-86 Limit Scaling（缩放极限）约束的参数面板

缩放极限约束和前面的位移极限、旋转极限约束设置类似，使用 Minimum 来设置坐标方向轴上的最小缩放比例，Maximum 为最大缩放比例。

#### 8.6.1.9 Maintain Volume（定积）约束

Maintain Volume（定积）约束可使骨骼在缩放变形时，自动控制坐标分量上的同步变形效果。如图 8-87 所示，为 Maintain Volume（定积）约束的参数面板。

**Free:** 物体体积的无约束控制坐标轴，其变形以本地坐标轴为准。

**Volume:** 体积的定积变化比例。

**Space:** 约束的变形空间。

首先添加一个球体物体，然后为其绑定一个骨骼物体。如图 8-88 左图所示，为未使用约束时，单击快捷键 S 将骨骼缩小 0.5 倍时的效果。接着为骨骼添加一个 Maintain Volume（定积）约束，并保持默认设置，这时再单击 S 缩小骨骼时，骨骼将保持其体积的定值，同步调整 XY 轴平面的形变，物体的缩放效果如图 8-88 右图所示。

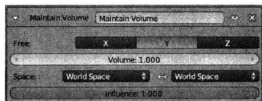


图 8-87 Maintain Volume（定积）约束的参数面板



图 8-88 使用 Maintain Volume（定积）约束的变形效果

#### 8.6.1.10 Transform（形变）约束

前面介绍的变形约束都是以相同类型的属性来控制被约束物体，那么 Transform（形变）约束则能使用不同属性的数据来约束被控物体的参数。如图 8-89 所示，为在一个立方体上添加一个 Transform（形变）约束后的参数面板。

**Extrapolate:** 外推模式，物体的约束范围在激活该选项后可被向外扩展，并被无限次地重复执行。

**Source:** 数据源，用于选择目标物体的数据类型。

**Destination:** 目标数据, 用于选择希望受被控物体影响的属性。

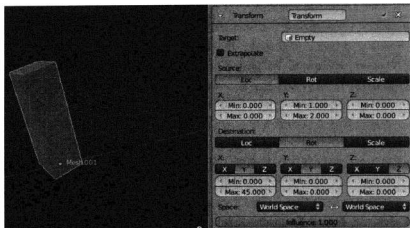


图 8-89 Transform (形变) 约束的参数面板

如图 8-89 所示, 希望使用目标物体的 Y 轴位移数据, 来约束被控物体在 X 方向轴上的转动角度, 因此首先在源输入中选择 Loc (位移) 的 Y 输入, 其中 Min 设置当前物体在 Y 轴上的最小位移距离, Max 为最大位移距离。接着在约束目标中, 选择约束为 Rot (旋转) 的 X 轴, 下面的 X/Y/Z 选项分别代表使用的输入源坐标轴, 这里可以选择 Y 轴, 调整 Min (最小值) 为物体转动的最小角度, Max 为转动的最大极限角度, 将其分别设置为 0 和 45 度。最后再选择 Empty (空) 物体, 并将其在 Y 轴上从单位坐标 1 移动至 2, 被控物体就会在 X 轴方向上执行从 0 到 45 度的等比例转动了。

## 8.6.2 跟随约束

### 8.6.2.1 Clamp To (紧固) 约束

Clamp To (紧固) 约束可通过调整物体的坐标数据, 来控制其在路径上的运动效果。如图 8-90 所示, 为 Clamp To (紧固) 约束的参数面板。

Clamp To (紧固) 约束和 Path Follow (路径) 约束效果类似, 都可将物体固定在一条指定路径上执行移动动画。它们之间唯一的区别就是: Path Follow 使用 F-Curve 曲线来控制物体的运动和速度, 而 Clamp To 则使物体在某一个约束方向上的移动, 来控制物体的运动和速度。例如, 使用如图 8-90 所示的设置, 将一个 Cube 物体绑定在 Curve 曲线路径上, 设置控制轴为 Z 轴。这时再选取 Cube 物体, 单击 G+Z 做 Z 轴上的垂直运动, 物体将根据其 Z 方向上的位移数据变化, 在曲线上产生同比例的路径动画效果, 如图 8-91 所示。而如果将物体沿 X 轴或者 Y 轴运动, 它将不会在曲线上产生任何移动效果。由于直观的控制方式, Clamp To 更适合制作曲线类的动画。

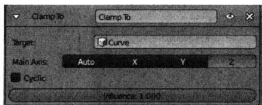


图 8-90 Clamp To (紧固) 约束的参数面板

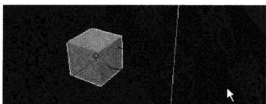


图 8-91 使用物体的 Z 轴运动来控制曲线动画的效果

### 8.6.2.2 Damped Track (衰减) 约束

Damped Track (衰减) 约束和 Track To (标准跟随) 约束功能类似, 区别在于: 后者没有逼近时的扭曲效果, 而前者在物体接近跟随物体的原点时, 会产生轻微的转动效果。如图 8-92 所示, 为

Damped Track（衰减）约束的参数面板，其中 to（目标）选项用于选择物体的转动坐标轴。

### 8.6.2.3 Inverse Kinematics（反向运动学）约束

Inverse Kinematics（反向运动学）约束是通过先确定子骨骼的位置，然后再反向计算其所在骨骼链上其他骨骼的联动情况，因此这是一种只能运用于骨骼的约束。在普通的骨骼运动中，关节的制作常采用 Forward Kinematics（前向运动学），例如要制作一个手臂动作，那么就会先调整上臂，然后是下臂的动作，最后再调整手掌。虽然这样调节的过程十分简单，运算速度也很快，但是每制作一个动作就需要调节所有的骨骼关节，需要花费大量的时间。而且由于各关节之间的内在关联性，直接设置各关节的形变值很容易产生不自然的动作。反向运动学的原理与此相反，它的基础就是依据子关节的位置和角度，来反向求解整个骨骼的形态。

反向运动学最大的优势就是工作效率高，可以有效地减少需要手动调节的骨骼数量；缺点在于它需要消耗较多的 CPU 来计算反向方程组，如果关节越多 IK 的运算将越慢。当完成骨骼的新建后，就可以切换至骨骼的 Pose（姿势）模式下，进入骨骼的约束菜单来为骨骼添加 IK 约束，如图 8-93 所示，为 Inverse Kinematics（反向运动学）的参数面板。

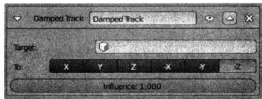


图 8-92 Damped Track（衰减）约束的参数面板

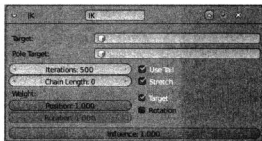


图 8-93 Inverse Kinematics（反向运动学）的参数面板

**Target:** 设置用于控制骨骼的目标物体，在同一骨架中常选择某一根特定的骨骼。

**Pole Target:** 设置用于控制骨骼扭转的目标物体。

**Iterations:** 设置骨骼的最大迭代数量。

**Chain Length:** 设置参与反向计算的最大骨骼数量。

**Use Tail:** 开启使用骨骼的尾部为控制原点。

**Stretch:** 开启骨骼的缩放效果。

**Weight:** 权重选项，分别调节反向计算对目标骨骼位移和转动的影响力度。

如图 8-94 左图所示，首先制作一个类似腿部结构的骨骼。如果要调整出一个弯腿的姿势，那么使用 FK 前向运算的方法，需要先转动大腿骨头 Bone，如图 8-94 中图所示，接着再转动小腿骨头 Bone.003，总共需要 2 次操作才能得到如图 8-94 右图所示的弯曲动作。

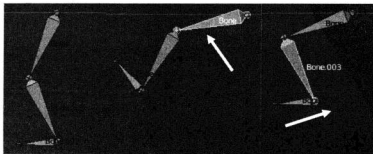


图 8-94 FK 前向运动控制制作的曲腿动作

接下来改用 IK 的方式,首先在 Pose (姿势) 模式下,单击 RMB 选择脚掌的骨头 Bone.002,为其添加一个 IK 约束控制,当前骨骼将在约束最远结点处的骨骼间产生一条虚线,用于标识建立的运算连接,如图 8-95 左图所示。由于脚掌离大腿有 2 个骨头,那么可以将反向距离 Chain Length 设置为 2,如图 8-95 中图所示。这时再选取脚掌骨头,单击 G 向 Z 上方移动,大腿和小腿骨头就会根据脚掌的运动自动计算转动的角度,如图 8-95 右图所示。可以发现,要达到同样的动作效果,使用 IK 控制只需要 1 次操作即可,而对比 FK 则需要使用 2 次以上的操作来完成。

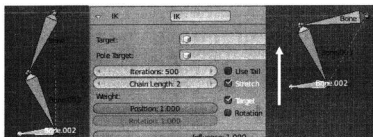


图 8-95 IK 反向运动制作的曲腿动作

#### 8.6.2.4 Locked Track (锁定跟随) 约束

Locked Track (锁定跟随) 约束,可用于制作一种类似指南针的跟踪效果。如图 8-96 所示,为 Locked Track (锁定跟随) 的参数面板。

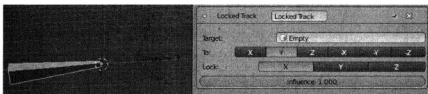


图 8-96 Locked Track (锁定跟随) 约束的参数面板

**Target:** 目标控制物体,这里输入用于跟踪的目标物体。

**To:** 选择跟踪目标物体的坐标轴数据源。

**Lock:** 选择将被锁定的坐标轴,这里要注意,Lock 中的锁定轴不能和 To 目标轴相同,否则将无法执行跟随计算了。如图 8-96 所示,将目标轴设置为 Y 轴,锁定轴为 X 轴,那么移动物体时,物体将只能在 YZ 平面上运动,并且时刻指向目标 Empty (空) 物体。

#### 8.6.2.5 Spline (样条曲线式) IK

Spline (样条曲线式) IK 是一种高级链式 IK 控制约束,如果要制作一条尾巴或鞭子等多骨骼的结构时,使用无约束的 FK 骨骼将很难完成平滑的链式动作,这里就可以使用 Spline IK 约束,利用较少的控制器来完成整个平滑动作的自动计算。如图 8-97 所示,为 Spline (样条曲线式) IK 的参数面板。

**Chain Length:** 用于设置链结构中的骨骼数量。

**Even Divisions:** 开启后,修改器将忽略骨骼长度与曲线的匹配性。

**Chain Offset:** 开启尾骨骼控制点与根骨骼的偏移量控制。

**Chain Scaling:** 其中 Y Stretch 用于调整骨骼上 Y 轴向的拉伸, XZ Scale Mode 则调整 XZ 平面上的骨骼拉伸,常用于将骨骼和曲线的弧度更完美地匹配起来。

**Use Curve Radius:** 启用曲线的 Radius (半径) 控制。

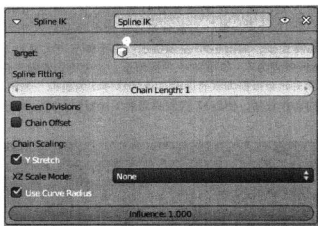


图 8-97 Spline（样条曲线式）IK 的参数面板

为了理解 Spline IK 的工作原理，首先在原心处新建一条 Path（路径），单击 Tab 进入其编辑模式，全选路径后按 G + X + 4 使其尾部移动至原心的位置，如图 8-98 左图所示。接着保持光标的位置，单击 Shift + A 添加一个 Bone（骨头），进入编辑模式，使用 Shift + D + G + X + 1 的命令，分别复制出 4 个同样的骨头。其中每个骨头都位于曲线的一个编辑节点位置上，将根部位置的骨头旋转 90 度，最后的效果如图 8-98 右图所示。

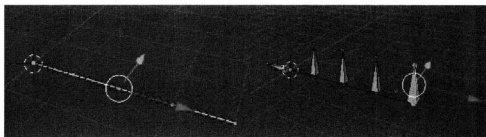


图 8-98 新建路径和控制骨头

接着单击 Tab 进入曲线的编辑模式，选择尾部的节点，为其添加一个 Hook 修改器，将 Object 控制物体设置为刚才新建骨骼物体中的单个 Bone。接着先单击 Reset（重置）Hook 的控制效果，然后再单击 Assign 将当前的点控制权赋予 Bone（骨头）。如图 8-99 所示，这时再进入骨头的 Pose 模式，便可以使用 Bone 来控制曲线当前尾部节点的运动了。

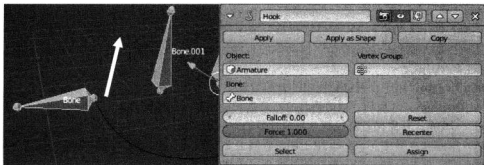


图 8-99 为曲线节点添加 Hook 控制



同理，曲线上一共有 5 个节点，分别为剩下的 4 个节点都添加 Hook 控制，如图 8-100 所示。这样就完成了曲线的骨头控制设计，只需要使用骨头即可控制整个曲线的形变了。

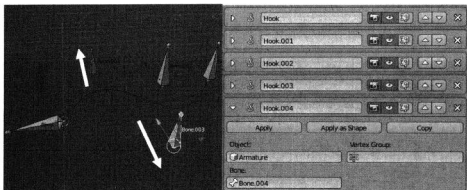


图 8-100 完成所有节点的 Hook 控制设置

为了将曲线的形变效果约束至网格物体上，接着在原点的位置添加一个新的骨骼物体 Armature.001，单击 Tab 进入其编辑模式后，将其头部移动至曲线的尾部位置，如图 8-101 左图所示。然后单击 W 对其做 3 次细分操作，最后的效果如图 8-101 右图所示。

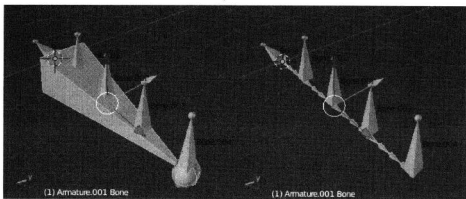


图 8-101 添加物体的网格控制骨骼

这个骨骼就是将要用于绑定网格的骨骼，它的形变将影响网格的变形效果。接着单击 Tab 进入其 Pose（姿势）模式，选择头部的骨骼，在约束面板中为其添加一个 Spline IK 控制器，如图 8-102 所示。设置 Target（控制目标）为曲线物体，接着将 Chain Length 修改为 18，关闭 Y Stretch 选项。

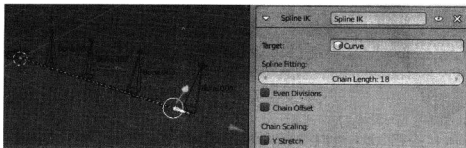


图 8-102 为骨头的头部添加 Spline IK 控制

这样，骨骼的链式动作将完全由 Spline IK 约束器来自动控制了。由于其形变是根据 Curve（曲线）物体的形变，为了将 Spline IK 骨头与控制曲线物体形变的骨骼 Armature 连接起来，选择 Spline 链式骨骼的尾部骨头，为其添加一个 Copy Rotation（转动复制）约束，设置其控制对象为曲线控制骨骼的第一节骨头 Bone，并设置约束坐标为 Y 轴，坐标空间为 Local Space（本地）坐标轴，如图 8-103 所示。

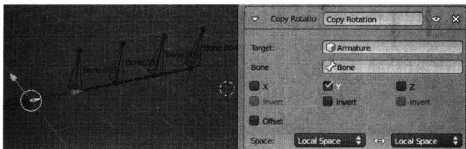


图 8-103 连接 Spline 骨骼与曲线骨骼的控制骨骼

为了演示最后的效果，添加一个网格物体，这里使用的是一个 Circle 物体，单击 E 将其扩展至骨骼的尾部，并使用 Ctrl + R 对其做多次细分，效果如图 8-104 所示。

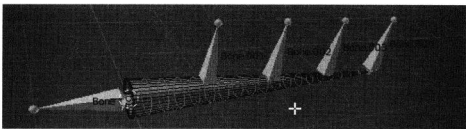


图 8-104 添加网格物体

首先选择网格物体，然后再单击 Shift + RMB 选择 Spline 骨骼，单击 Ctrl + P 建立父子关系，并设置 With Automatic Weights（自动权重计算）模式，如图 8-105 左图所示。现在只需要使用曲线的控制骨骼，即可完成对整个尾巴物体的链式动作调整了，十分方便、快捷，其中骨骼间的弯曲和旋转效果将由 Spline IK 系统来自动计算。绑定的最后效果如图 8-105 右图所示。

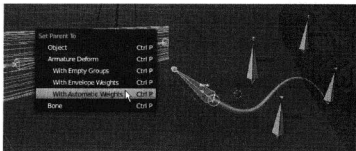


图 8-105 完成物体与骨头的绑定

#### 8.6.2.6 Stretch To（拉伸追踪）约束

Stretch To（拉伸追踪）约束使得物体在对目标做跟踪运动的同时，可以根据物体与目标之间的距离产生相应的拉伸形变。如图 8-106 所示，为 Stretch To（拉伸追踪）约束的参数面板。

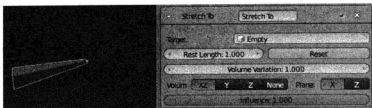


图 8-106 Stretch To (拉伸追踪) 约束的参数面板

**Target:** 选择目标控制物体。

**Rest Length:** 物体原心到目标物体间的距离会在 Rest Length 中显示出来, 可以使用 Reset 来重新计算物体间的距离。Rest Length 是物体形变的比例基准, 物体将根据这个距离的变化来做等比例的拉伸。

**Volume Variation:** 体积变化率, 用于控制物体的拉伸比例, 当值为 1 时缩放的影响比例为 100%。这时当距离增大 20% 时, 物体也将被拉伸 20%。

**Volum:** 选择被拉伸的物体形变坐标轴。

**Plane:** 选择形变中被固定的坐标面。

#### 8.6.2.7 Track To (标准跟随) 约束

Track To (标准跟随) 约束能提供一个完全跟踪的约束效果, 常用于骨骼设计中, 与 IK 约束配合使用。如图 8-107 所示, 为 Track To (标准跟随) 约束的参数面板。



图 8-107 Track To (标准跟随) 约束的参数面板

**To:** 选择物体指向目标物体的坐标轴。

**Up:** 物体被约束的坐标轴方向, 当激活 Target Z 时, 物体将使用目标物体的 Z 轴为约束旋转轴。

### 8.6.3 关系约束

#### 8.6.3.1 Action (动作) 约束

Action (动作) 约束常用于骨骼中, 将目标骨骼的动作映射到被约束骨骼上。如图 8-108 所示, 为 Action (动作) 约束的参数面板。

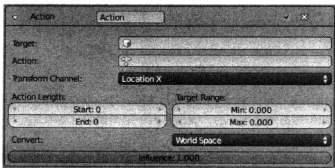


图 8-108 Action (动作) 约束的参数面板

**Target:** 输入目标骨骼的名称。

**Action:** 设置被调用的骨骼 Action (动作) 名。

**Transform Channel:** 选择被映射的坐标轴和动作类型, 这里可以选择 Location (位移)、Rotation (转动) 和 Scale (缩放) 类型的动作。

**Action Length:** Start 和 End 分别设置动作的起始和结束帧位置。

**Target Range:** 用于控制动作对目标骨骼通道的影响范围。

### 8.6.3.2 Child Of (父子) 约束

除了使用快捷键 P 来建立父子关系外, 还可以使用约束控制来建立父子关系。如图 8-109 所示, 为 Child Of (父子) 关系约束的参数面板。

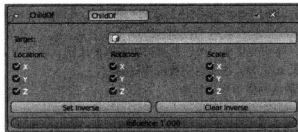


图 8-109 Child Of (父子) 关系约束的参数面板

**Target:** 设置为被约束物体的父物体, 这里与物体模式中建立父子关系的方式不同, 约束修改器提供了更严谨的设置, 允许自定义建立 Location (位移)、Rotation (转动) 和 Scale (缩放) 通道的父子关系。

**Set Inverse:** 保持子物体与父物体之间的形变偏移量。

**Clear Inverse:** 清除父子物体间的偏移量。

### 8.6.3.3 Floor (地面) 约束

Floor (地面) 约束可为物体建立一个地面或墙面约束, 禁止物体的原心从其中穿过。如图 8-110 所示, 为 Floor (地面) 约束的参数面板。



图 8-110 Floor (地面) 约束的参数面板

**Target:** 设置目标物体, 也就是作为地面的物体名称。

**Sticky:** 激活时物体将不能在地面上产生滑动效果, 这一点在制作人物行走时十分有用。

**Use Rotation:** 如果地面不是水平的, 而是具有一定倾斜的, 激活 Use Rotation 将自动计算地面的角度。

**Offset:** 偏移量, 用于设置地面与骨骼原心的距离偏移量, 例如骨骼的原心垂直距离地面为 0.5 个单位, 那么设置偏移 0.5 可让骨骼在站立时, 底部刚好接触到地面。

**Min/Max:** 用于定义地面的坐标系, 例如将物体设置为一堵墙时, 可定义其为 Z。

#### 8.6.3.4 Follow Path (路径跟随) 约束

Follow Path (路径跟随) 约束可将物体的运动约束在一个曲线物体上, 如图 8-111 所示, 为 Follow Path (路径跟随) 约束的参数面板。

**Target:** 选择目标曲线物体, 用于定义物体的运动路径。

**Follow Curve:** 是否允许物体跟随路径弯曲的方向做同步转动。

**Curve Radius:** 是否允许物体根据曲线上点的 Radius (半径值) 做缩放变化。

**Fixed Position:** 将物体固定在路径上的某一点, 其中 Offset 偏移量用于控制物体在曲线上的位置。

**Forward:** 设置曲线上物体的运动坐标方向。

#### 8.6.3.5 Pivot (转轴) 约束

Pivot (转轴) 约束可使骨骼或物体能自定义旋转原点, 如图 8-112 所示, 为 Pivot (转轴) 约束的参数面板。

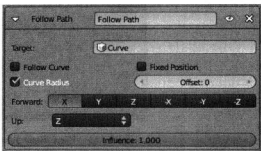


图 8-111 Follow Path(路径跟随) 约束的参数面板

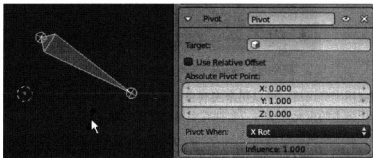


图 8-112 Pivot (转轴) 约束的参数面板

在一根骨骼上添加一个 Pivot (转轴) 约束, 修改 Y 轴上的转轴距离为骨骼长度, 并将转轴的坐标锁定为 X 轴。这样在 Pose 模式下使用快捷键 R + X 旋转骨骼时, 骨骼将以骨骼的头部位置, 也就是重新定义的 Pivot 点为转轴做旋转运动。

**Target:** 使用物体来定义新的转轴位置。

**Use Relative Offset:** 启用目标物体的相对偏移量设置。

**Absolute Pivot Point:** 手动定义转轴的坐标位置。

**Pivot When:** 定义转动的坐标约束轴。

#### 8.6.3.6 Rigid Body Joint (铰链) 约束

Rigid Body Joint (铰链) 约束, 常用于游戏引擎中来模拟链式效应。如图 8-113 所示, 为 Rigid Body Joint (铰链) 约束的参数面板。

**Pivot Type:** 选择物体间的链接方式。

**Child Object:** 设置与当前物体链接的子物体。

**No Collision:** 可开启物体间的自碰撞效果。

**Display Pivot:** 开启物体转轴上的坐标轴显示。

**Pivot:** 控制转轴在物体上的偏移位置。

**Axis:** 调整转轴在各坐标轴上的转动角度。

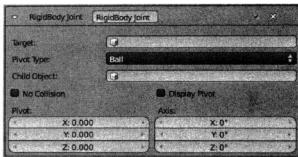


图 8-113 Rigid Body Joint (铰链) 约束的参数面板

#### 8.6.3.7 Script (脚本) 约束

Script (脚本) 约束可以为物体或骨骼添加使用 Python 脚本编写的约束控制, 如图 8-114 所示, 为 Script (脚本) 约束的参数面板。



图 8-114 Script (脚本) 约束的参数面板

脚本约束的应用十分灵活, 可以用来实现自定义的约束需求。例如常规的约束只允许一个目标物体, 而使用脚本约束允许编写一个拥有多个目标物体的约束。

#### 8.6.3.8 Shrinkwrap (收缩) 约束

Shrinkwrap (收缩) 约束用于将物体或骨骼约束在物体的表面上, 如 8-115 所示, 为 Shrinkwrap (收缩) 约束的参数面板。

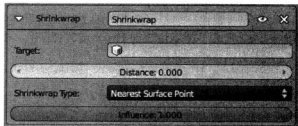


图 8-115 Shrinkwrap (收缩) 约束的参数面板

**Distance:** 定义物体原心与网格表面的距离偏移量。

**Shrinkwrap Type:** 选择收缩约束的模式, 其中 Nearest Vertex 将约束至离目标物体最近的网格点上, Project 约束在对应的坐标轴上, Nearest Surface Point 则可以约束至最近的网格表面上。

## 物理特效与仿真

在CG制作中,常使用系统集成的物理引擎,模拟一些真实重力下物体间相互作用的效果,让虚拟环境可以更逼真地还原真实的自然世界,制作出令人信服的动画效果。

### 9.1 粒子仿真系统

Particle(粒子)系统是一种特殊的模糊现象仿真,常用于模拟毛发和烟火等的效果,以及蜂群等大量的集群体动画。粒子系统的运动和位置状态是由发射器来控制的,而粒子的行为主要由粒子的初始速度、初速度向量、粒子寿命和外力影响等参数来控制。

在Blender中,每一个物体可搭载多个粒子发生器,每个粒子发生器可最多产生100 000个粒子,同时每个粒子还可扩展出10 000个子粒子,这么多的粒子数量完全可以满足设计需求。但是粒子模拟需要消耗大量的内存,场景的承受能力也依赖于系统的内存容量,所以粒子的数量越多,预览和渲染的速度也将越慢,当粒子数量超过了系统的承受力时,Blender就可能崩溃。

粒子的制作需要经历模拟阶段和渲染阶段。在模拟阶段,系统会根据粒子的初始速度和新生间隔来计算场景中将要产生的粒子数量,如果粒子在场景中的存在时间超过了其生命周期,就会从模拟过程中被剔除掉,称此时粒子被系统执行了回收操作。粒子的物理模拟包括了对自身特性执行的运算,以及对粒子间的碰撞检查和与障碍物之间的反弹计算。如图9-1所示,为电影《Big Buck Bunny》中,使用粒子制作的雄兔毛发和草坪效果。



图9-1 《Big Buck Bunny》中使用粒子工具制作的雄兔毛发和草坪效果

当完成了粒子的模拟，系统就会将模拟的数据保存在本地磁盘上。这时只需要为粒子的发射物体添加合适的材质和纹理，即可单击 F12 由系统重新读取存储的模拟数据，并执行渲染计算了。这里将粒子的制作流程归纳如下：

- 1) 制作用于承担粒子发生器角色的物体模型，这个物体不一定需要在渲染中可见，但是它的着色器属性能影响粒子的颜色等可视效果。
- 2) 在物体网格上建立 Particle System（粒子系统），并设置粒子发生器的相关属性。
- 3) 调节粒子的行为属性，例如毛发的长度等，并添加粒子与环境的碰撞参数。
- 4) 为网格物体模型制作全局动画等动作，并设置外力物体的属性，例如 Wind（风）等。
- 5) 在物体模式下运行模拟烘焙，最后再单击 F12 渲染结果。

### 9.1.1 粒子系统

如图 9-2 所示，可以在属性面板中找到粒子系统菜单，单击旁边的十字星箭头，为当前物体添加一个新的粒子系统。

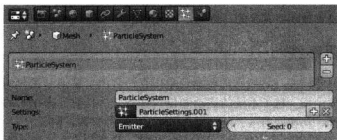


图 9-2 新建粒子系统的参数面板

**Name:** 粒子系统的命名与重命名选项。

**Settings:** 可以调用场景中另一个粒子系统的参数设置。

**Seed:** 粒子的发射偏移量，可为粒子提供一个随机的发射源。

**Type:** 粒子类型，这里包含了常用的粒子类型，常用的有 Emitter（发射）类和 Hair（毛发）类。

第一种 Emitter（发射）类系统常被用于制作运动的粒子效果，当搭配烟雾等物理模拟时，可制作火焰的场景。如图 9-3 所示，为 Andrew Price 制作的篝火场景。

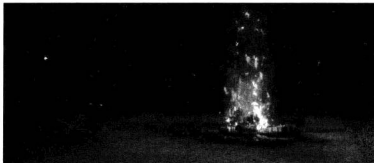


图 9-3 Andrew Price—Realistic Fire

第二种为 Hair（毛发）类粒子，常用于制作头发和草坪等效果。作为静态类的粒子物体，系统增加了 Segments（分段）设置，用于控制毛发的平滑细分等级。同时 Emission（发射器）面



板中还取消了粒子发射周期和生命周期的设置,使粒子直接利用其运动路径来模拟毛发的长度。如图9-4所示,为Hair模式下的Emission(发射器)参数面板。

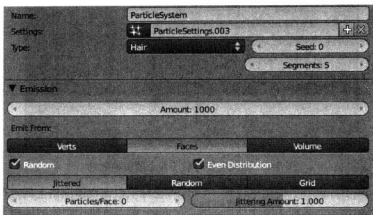


图9-4 Hair模式下的Emission(发射器)参数面板

在使用Hair制作毛发等动画时,应尽可能地降低Amount(数量值),特别是当需要为毛发添加软体模拟等物理特性时。一般来说,Amount设置为2000是一个比较理想的经验值,这能在一个足够理想的可视效果内实现最优化的性能,而毛发的层次和细节可以使用粒子来添加。Segments(分段细分)也是影响毛发物理计算的另一个重要参数,因为细分等级越高,每一根毛发上的控制点也就越多,软体计算中对点的运算量会因此成倍增加。所以对于长发来说,Segments设置为5就足够了;而对于短发来说,设置为2到3即可。

### 9.1.2 发射器选项

顾名思义,Emitter(发射器)可控制粒子像子弹一样在法线方向上发射出去,如图9-5所示,为Emission(发射器)的参数面板。

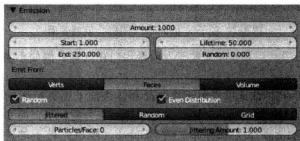


图9-5 Emission(发射器)的参数面板

**Amount:** 用于控制当前粒子系统的主粒子数量,粒子数量越多,消耗的内存也越多,所以这里应根据场景选择合适的粒子数。过大的数量设置不一定能带来更好的效果,反而会加重渲染的负担。由于可以使用粒子来丰富粒子细节,因此这里的数字并不包括粒子的数量。

**Start/End:** 粒子发射的起始时间帧位置和结束帧位置。

**Lifetime:** 粒子的生命周期,时间越长粒子在场景中存活的时间越久,反之将消失得越快。

**Random:** 为粒子的生命周期添加一定的随机系数,当系数为0时,粒子将按照指定的时间点被系统回收掉,一旦增大系数设置,粒子的寿命会产生一定的随机变化。例如,初始化为50帧的粒子寿命,如果设置随机系数为0.5,那么粒子寿命将会有  $50 \times (1 - 0.5)$  帧的随机波动范

围。如图 9-6 左图所示，为没有系数影响寿命的粒子效果图，而右图为添加系数后，粒子发射器上产生粒子获得不同随机寿命的效果。

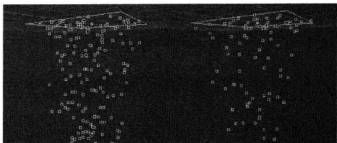


图 9-6 随机系数对粒子寿命的影响效果

**Emit From:** 用于控制粒子的出生位置，Verts 可使粒子根据物体网格上的点位置来计算出生地，而 Faces 则将会使用面来作为粒子的出生地，Volume 选项将使用物体的体积为计算标准。

**Random:** 激活后粒子将按照随机的顺序从出生地向外发射。如图 9-7 左图所示，为未激活 Random 时粒子的发射效果，可以看到，粒子按照网格的区间区域向外发射，而右图则是激活 Random 选项后随机发射粒子的效果。

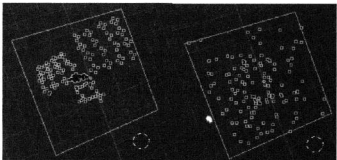


图 9-7 粒子的 Random（随机）发射效果

**Even Distribution:** 激活后系统将对粒子发射做均匀化处理，例如小密度的点面区域将产生较少数量的粒子。

**Jittered/Random/Grid:** 该选项仅用于修改面和体表面上运算粒子的发射分配顺序，其中 Jittered 将产生抖动的粒子效果，Random 则使用随机顺序来分配粒子，而 Grid 会使用网格的点阵顺序来产生粒子物体。如图 9-8 左图所示，为 Random（随机）粒子的发射效果，右图为 Grid（点阵）顺序发射粒子的效果。

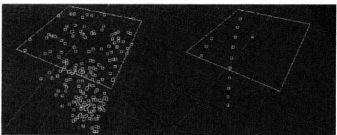


图 9-8 粒子的发射分配顺序

### 9.1.3 物理效果选项

粒子的运动效果由物理系统来控制,如图 9-9 所示,为粒子系统的 Physics (物理) 选项参数面板。

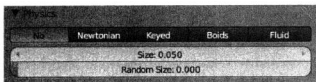


图 9-9 Physics (物理) 选项的参数面板

当选择 No (无) 物理效果时,粒子将不执行任何物理模拟运算,粒子系统也将失去动态效果。但是关闭物理模拟并不代表粒子系统会失效,场景中新产生的粒子将在生命周期内,紧贴在发射器物体的表面上,直到寿命期结束时被回收掉。

#### 9.1.3.1 Newtonian (牛顿力学) 仿真

Newtonian (牛顿力学) 仿真是最常用的物理模拟计算方式,如图 9-10 所示,为 Newtonian (牛顿力学) 仿真的参数面板。

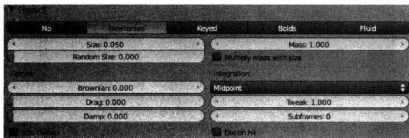


图 9-10 Newtonian (牛顿力学) 仿真的参数面板

**Size:** 粒子尺寸设置,其中的 Random Size (随机尺寸) 选项可为粒子添加一定的随机尺寸效果。

**Forces:** 外力效应,使用 Brownian (布朗动力学) 系数来为粒子之间添加不规则的运动力,特别是在较微弱外力下的模拟时,可适当增大系数来获得明显的外力效果。

**Drag:** 拖拽系数,使系统可根据粒子的当前速度和尺寸来模拟环境对粒子的阻力,常用于模拟空气和水等对物体的阻力效果。

**Damp:** 阻力系数,用于调节物体与摩擦时对粒子速度产生的影响效果。

**Mass:** 粒子的平均质量参数,用于调节粒子的质量,其中 Multiply mass with size 选项可让系统根据粒子的质量自适应地调整粒子尺寸。

**Size Deflect:** 开启粒子的尺寸偏转量计算。

**Die on hit:** 允许粒子在碰撞后可以被相互吸收。

**Integration:** 粒子的运动路径计算采用了数学的积分器来执行控制与模拟,这里提供了 4 种积分运算模式。

- ☐ Euler 欧拉法是一种常用的积分算法,也是最简单最快速的计算方式,但是它的缺点是很不稳定,能量在无阻力系数时不会随时间的推移而衰减。因此这种模式最适合快速的预览仿真,或者对精度要求较低的无阻尼场景。
- ☐ Verlet 综合化算法,是由法国著名物理学家 Loup Verlet 根据牛顿定理提出的物理计算方

程，它的计算过程比 Euler 算法更简单，并且拥有更准确的碰撞运算结果，常用于流体仿真。

- ❑ Midpoint 中值法，它的计算速度比 Euler 算法慢，但是更稳定。如果模拟的加速度为常值，那么运动的能量效果将守恒，因此一般采用 Midpoint 为默认的分器。
- ❑ RK4 是四阶龙格库塔算法（4th Order Runge-Kutta）的缩写，它和 Midpoint 算法类似，但后者更类似于一种二阶的龙格库塔算法。RK4 的计算速度比 Midpoint 算法更慢，但是精度却大幅提升，可以在非定加速度的情况下保持能量平衡。因此，在时间和计算量允许的范围，可以使用 RK4 算法来替代 Midpoint 算法，完成更复杂的模拟。

### 9.1.3.2 Keyed（关键体）仿真

如图 9-11 所示，为 Keyed（关键体）仿真的参数面板。

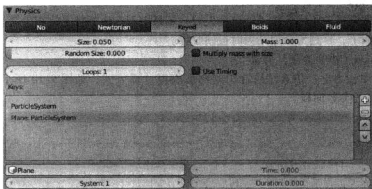


图 9-11 Keyed（关键体）仿真的参数面板

Keyed（关键体）仿真可以将多个粒子系统构成一套链式系统，用于制作特殊的静态或动态粒子效果，例如光纤或纺织动画等。如图 9-12 所示，为一个使用图 9-11 设置的静态 Keyed 仿真效果，可以看到球体上发射出的粒子根据目标 Plane 物体上粒子系统的设置，被全部牵引至 Plane 物体的表面上。

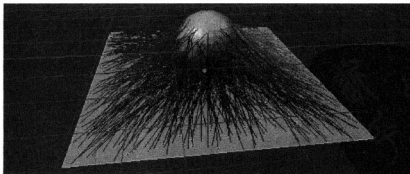


图 9-12 静态关键体粒子的仿真效果

要实现链式粒子的效果，需要在 Keys 列表中添加至少两组以上的粒子系统，系统将会根据列表中关键体的顺序，自动计算粒子在物体之间的运动路线。每一个 Keyed（关键体）允许使用独立的粒子系统，可以为其分别设置与发射源不同的物理运算模式。在 System（系统）选项中选择当前关键体将要使用的粒子系统，如果选择的系统 Physics（物理）设置为 No，那么关键体将只产生一种类似于纯发射源的粒子效果。

列表中的第一个 Keyed 作为发射源关键体，排列在第一层，所有的粒子将以此物体为起始位置向下面的 Keyed 依次传递，最下面的 Keyed 将为粒子的目标体。如果不指定一个目标物体，粒子将默认吸附在 Keyed 物体本身的表面上。

继续添加一个新的目标物体 Plane.001，并排列在 Plane 之后，如图 9-13 所示，粒子将在链接第一个物体后，自动与第二个物体构成链式的静态粒子效果。



图 9-13 多目标的 Keyed 仿真效果

对于使用 Keyed（关键体）制作动态的粒子效果，可将粒子从源物体依次传递至链上的其他关键物体上。如图 9-14 所示，新建一个 Cube，为其添加一个 Keyed 粒子系统。接着在 Keys 列表中新建 3 个独立粒子系统物体，分别命名为 A、B 和 C，并将它们的 Physics 设置为 No。单击快捷键 Alt + A 运行模拟后可以发现，由于 Cube 上的粒子系统不在 Keys 列表中，所以粒子效果将不会作用在其表面上。它的粒子效果将按照列表的顺序，从 A 发射到 B，然后再发射至 C。如果启动了 Loops（循环）模式，粒子将在到达 C 物体后返回到 A，做重复的循环动态粒子运动。

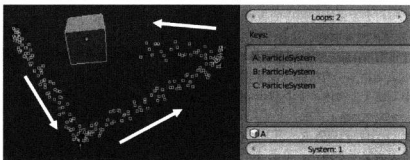


图 9-14 多关键体的粒子循环运动效果

可以通过修改粒子的生命周期等参数，来调节关键体间粒子的运动速度。对于同样的仿真周期，更短的粒子寿命意味着更快的粒子运动，同时关键体数量越多，粒子的运动速度也将越快。但是这样设置的缺点就是：粒子的速度在各个系统间都是平均的，如果要定义不同系统间的速度，可以开启 Use Timing（时间线）选项。

#### 9.1.3.3 Boids（群体）仿真

Boids（群体）仿真是一种可以模拟鸟群或蜂群等大规模集群物体运动的粒子仿真模式，它将使用粒子来替代集群中的单个物体，并使粒子之间可按照群体的行为规则产生交互的作用效果。如图 9-15 所示，为 Boids（群体）仿真的参数面板。

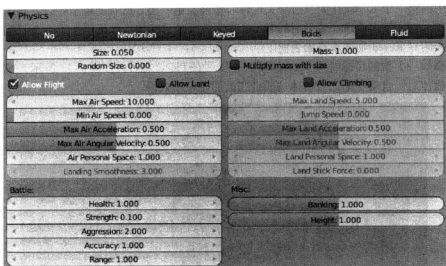


图 9-15 Boids（群体）仿真的参数面板

**Size/Mass:** 控制粒子的基本属性，如尺寸和质量等。

**Max/Min Air Speed:** 当开启 Allow Fight（飞行）模式时，需要定义空气速率的取值范围，场景中的平均空气速率将取两个极值的平均值。

**Max Air Acceleration:** 空气的最大加速度系数，用于控制物体的直线加速度。

**Max Air Angular Velocity:** 空气的最大角速度，用于控制物体的转向加速度。

**Air Personal Space:** 物体在空中的空间尺寸，用于设置碰撞的检测边界。

**Allow Climbing:** 是否允许开启地面模式，这时粒子将不再在空气中飞行，而是直接在地面上前进，并按照行为逻辑执行相应的运动。

**Battle:** 物体的战斗属性，这常应用在一些特殊的动画效果中，例如两军交战的场面。

**Misc:** 这里的 Banking 用于控制物体转弯时的倾斜度，默认值为 1 时是最自然的倾斜效果，而 Height 高度值则用于调整物体与粒子的高度尺寸比例。

在场景中新建 3 个物体：一个 Plane（平面）物体、一个自定义球体和一个大圣物体。首先为 Plane 物体添加一个粒子系统，并将粒子的物理模式修改为 Boids（群体）模式，把粒子发射数修改为 400，发射帧数设置为从第 1 帧到第 3 帧，粒子的生命周期调整为 400，将粒子的显示方式替换为自定义的球体物体，单击 Alt + A，粒子的发射效果如图 9-16 所示。此时可以看到，球体将产生无规律的随机运动效果，并缓慢地向四周散开。

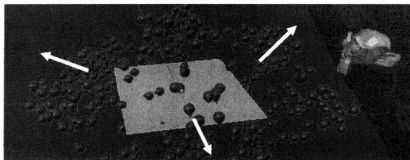


图 9-16 无规律的 Boids（群体）运动效果

当为 Plane 物体设置了 Boids 属性后,面板中会弹出一个新的 Boid Brain 逻辑面板,可以在这里为群体物体设计行为逻辑,如图 9-17 所示。

单击旁边的十字箭头,即可在列表中添加一个 Goal (目标) 逻辑,将 Goal 物体设置为大圣。由于列表中的运算逻辑是从上至下的,所以这里需要单击移动箭头将 Goal 移至第二位。这样系统将首先执行 Separate (分散) 逻辑,将粒子物体之间相互分离开,然后再根据 Goal (目标) 逻辑计算粒子物体的跟随路径,最后再将路径分配给各物体执行群集的 Flock (聚集跟踪) 逻辑,使所有的粒子都朝向目标物体运动,运算效果如图 9-18 所示。

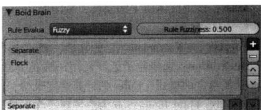


图 9-17 Boid Brain 行为逻辑面板

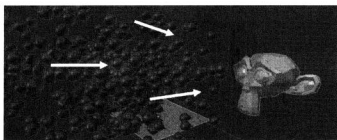


图 9-18 Boids 的跟随运动效果

#### 9.1.3.4 Fluid (流体效果) 仿真

Fluid (流体效果) 仿真是 Blender 使用粒子来完成流体仿真的一种方式,采用的是 SPH (Smoothed Particle Hydrodynamics, 光滑粒子流体动力学) 仿真模式。

SPH 是以流体中各个质点的速度和密度等属性参数为基础,在流体质点的运动过程中,控制其属性参数值随时间的变化效果,从而构成了整个粒子流体的效果仿真。它的基本特点是使粒子系统能追踪流体质点的运动,同时可直接运用固体力学中的质点动力学来参与仿真计算。由于 SPH 是一种纯 Lagrange (拉格朗日) 的粒子流体计算方法,能避免 Euler (欧拉法) 中欧拉网格与材料的接触面问题,因此特别适合于完成高速碰撞等大动态的流体场景模拟。如图 9-19 所示,为 SPH 流体的参数面板。

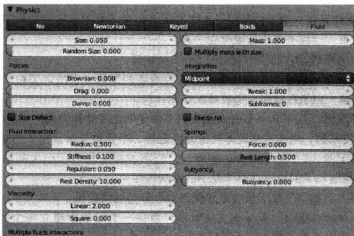


图 9-19 SPH 流体的参数面板

**Size/Random Size:** 粒子的尺寸控制以及尺寸的随机系数设置。

**Mass:** 设置流体粒子的质量, 激活 *Multiply mass with size* 可使系统根据粒子的质量自适应地调整粒子尺寸。

**Forces:** 与 *Newtonian* (牛顿物理) 模式中的外力效应参数原理类似。

**Integration:** 选择粒子运算使用的插值算法, 合适的选择可使模拟更稳定。一般来说 *Mid-point* 是最稳定的模式, *Euler* 的计算速度最快, 而 *Verlet* 是最符合流体行为运动学的插值算法。

**Radius:** *Fluid Interaction* 为流体粒子在碰撞交互时的作用力参数选项, 用于控制流体之间的相互作用力效果。其中 *Radius* 半径是流体粒子中最重要的控制参数, 用于设置粒子的反弹作用力半径, 它决定了粒子仿真对象的物理属性。例如, 半径值越小的粒子间反弹作用力越弱, 常用于制作大尺寸的流体 (如沙粒和雪等), 其参数可设置在 0.1 至 0.2 之间。而对于小尺寸的粒子物体 (如水滴等物体), 它们之间的作用力较强, 则应该使用较大的半径数值, 一般设置在 1.0 以上。如图 9-20 左图所示, 为一个简单场景, 当设置 *Radius* 为 0.25 时, 可以看到粒子间的反弹作用力较弱, 很容易聚合在一起。而右图则为将 *Radius* 增大至 0.75 时的效果, 可以看到粒子的发散性变强了。

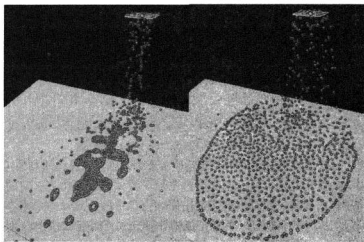


图 9-20 不同 *Radius* 粒子作用力半径的仿真效果

**Stiffness:** 抗扰性系数, 用于控制粒子表面的张力效果, 对于小尺寸的粒子常使用较大的参数来产生大张力效果, 使粒子更容易聚合在一起。

**Repulsion:** 斥力系数, 它的作用效果与 *Stiffness* 相反。

**Rest Density:** 密度重置系数, 这也是一个十分重要的参数, 它决定了粒子在二次碰撞后的密度恢复能力, 值越大粒子在碰撞后越容易汇聚在一起。

**Springs:** 弹性参数, 其中 *Rest Length* 用于调节粒子间由于弹性力而产生的距离变化比例。

**Viscosity:** 黏度系数, 用于控制流体粒子的稳定性, 当值越低时, 粒子的黏度越强, 越容易聚合。这里应区别于 *Force* 中的 *Damping* 参数, 它们两者都能使粒子产生高粘度的效果, 但是 *Damping* 主要用于控制粒子的速度, 而非粒子的本身属性。

**Buoyancy:** 浮力系数, 为粒子运动添加受浮力上升的运动效果。

**Multiple fluid interactions:** 多流体交互效果, 当在一个场景中建立多个 SPH 粒子系统时, 可分别添加互动作用力解算, 使各粒子系统间无法产生混合效果。如图 9-21 所示, 在场景中添加两个 SPH 粒子系统, 并分别在 *Multiple fluid interactions* 选项里添加对方粒子系统的名称, 单击



Alt + A 后可以发现,产生的粒子由于无法混合而在交界处实现了聚集的交互现象。

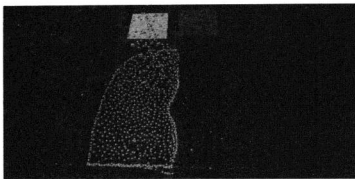


图 9-21 Multiple fluid interactions 多流体交互的仿真效果

SPH 粒子模拟的参数众多,它们之间的效果可相互叠加,但是这种重叠结果不是简单的求和计算,而是一种非线性的组合运算结果。所以掌握它们不仅需要大量的实验和摸索,还需要了解流体运动的一些基本原理。具体有关流体的运动学仿真,将在后面着重讨论。

#### 9.1.4 可视化选项

Display (可视化) 面板能控制粒子在场景中的显示情况和渲染时的可视效果,如图 9-22 所示,为粒子的 Display (可视化) 选项的参数面板。

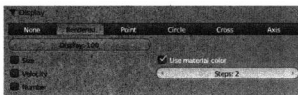


图 9-22 Display (可视化) 选项的参数面板

**Display:** 可视化面板提供了 6 种粒子的场景显示模式,其中 Display (显示) 选项能调整场景中粒子的可视化显示比例,例如当设置比例参数为 100 时,100% 数量的粒子都将被显示在场景中。如图 9-23 所示,从左至右分别为 Point、Circle、Cross 和 Axis 的粒子显示效果。

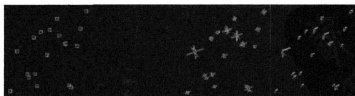


图 9-23 粒子的不同显示查看方式

**Size/Velocity/Number:** 分别开启粒子的实际尺寸显示、速度矢量显示和粒子编号显示。

**Use material color:** 能使用当前发射体上赋予的材质颜色来显示粒子,取消后将默认粒子颜色显示为黑色。

**Steps:** 定义粒子或毛发在渲染时的细分等级,默认为 2。

#### 9.1.5 渲染选项

在 Render (渲染) 面板中,可以定义粒子在渲染时的可视效果。如图 9-24 所示,为 Render

(渲染)的参数面板。

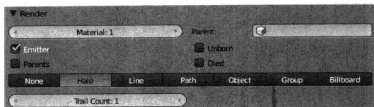


图 9-24 Render (渲染) 的参数面板

**Material:** 一个物体能搭载多个粒子系统, 可以在 Render 面板中为每个粒子系统分别赋予单独的材质着色器, 只需要在 Material 中选择材质序列即可。

**Parent:** 父坐标选项, 用于重定义粒子的发生坐标系。在这里输入的物体将使用其本地坐标, 来替换当前粒子系统的全局坐标。如图 9-25 所示, 为使用 Empty 物体的本地坐标来重新定位粒子发生坐标系的效果。

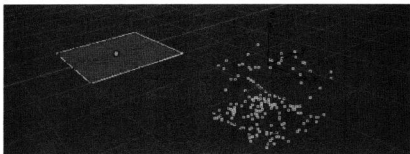


图 9-25 Parent 重定义粒子产生的坐标系

**Emitter:** 如果不希望最后渲染出粒子的发生物体, 可以取消 Emitter (发射器) 选项。

**Parents:** 可在渲染粒子粒子的同时, 开启父粒子的渲染。

**Unborn/Died:** 分别控制超出粒子生命周期的渲染方式, Unborn 能渲染未出生的粒子, 而 Died 将渲染生命周期已结束的粒子。

**Trail Count:** 可在渲染时为粒子添加一定数量的尾随粒子效果, 如图 9-26 左图所示为默认的粒子效果, 当适当地增大 Trail Count 参数时, 场景中的粒子将在原运动路径上产生尾随的新粒子, 效果如图 9-26 右图所示。

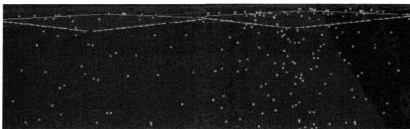


图 9-26 Trail Count (尾随粒子) 的仿真效果

Blender 提供了 6 种渲染的显示方式, 当选择为 None 时, 当前粒子将不会被渲染出来。系统默认设置为 Halo (光晕) 选项, 并采用 Halo (光晕) 材质来渲染粒子。

### 9.1.5.1 Line (线条) 模式

Line (线条) 模式下, 粒子将被渲染为线段效果, 如图 9-27 所示。其中 Back 和 Head 将分别控制线段的尾部长度和头部长度, Speed 选项能根据粒子的速度来调整线段的长度。

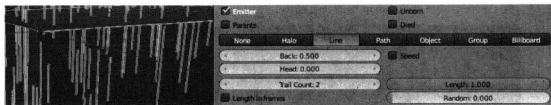


图 9-27 Line (线条) 模式

### 9.1.5.2 Path (路径) 模式

Path 模式适用于烘焙的粒子和 Keyed (关键帧) 模拟下的粒子效果, 常用于毛发与草丛的渲染。如图 9-28 所示, 为一个静态 Keyed (关键帧) 的粒子效果。

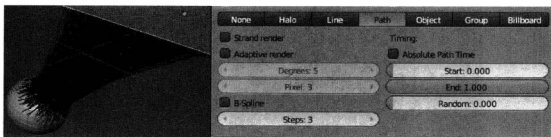


图 9-28 Path (路径) 模式

**Strand render:** 开启粒子的静态线股渲染。

**Adaptive render:** 自适应渲染模式, 能在渲染前将路径上不必要的几何体移除, 以降低内存消耗, 同时减少渲染的时间。

**B-Spline:** 开启 B 样条插值模式来计算粒子路径, 其中 Steps 可设置路径的细分等级, 细分等级越高, 线条越平滑。

**Timing:** 根据时间来控制渲染的路径长度, 而默认是使用比例值来控制。当 Absolute Path Time 被激活后, 系统可根据帧的数值设置来精确控制时间点。

### 9.1.5.3 Object (物体) 模式

Object 模式下, 系统会在渲染时, 将粒子替换为在 Dupl Object 中输入的物体。如图 9-29 所示, 新建一个 Drop (锥形) 物体, 并将其设置为替换物体, 粒子在渲染时就被全部替换为 Drop 物体了。如果激活 Use Global 选项, 粒子将使用替换物体的坐标轴为粒子源。



图 9-29 Object (物体) 模式

#### 9.1.5.4 Group (群组) 物体模式

Group 模式和 Object 模式类似, 只是将替换物体从单个物体更改为群组的物体。如图 9-30 所示, 将一个柱体和椎体使用快捷键 Ctrl + G 加入组 Group 中, 然后在 Dupli Group 中输入组 Group 名称, 粒子就被自动替换成 Group 中的所有物体。



图 9-30 Group (群组) 物体模式

**Whole Group:** 将粒子效果同时应用在每一个替换物体上, 如图 9-31 所示。

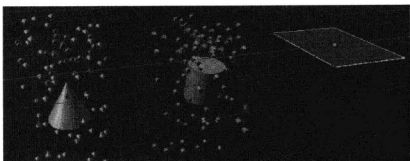


图 9-31 Whole Group 的仿真效果

**Use Global:** 分别使用 Group 中的物体本地坐标来控制粒子源。

**Use Count:** 可分别控制组中物体在粒子中的显示比例, 如图 9-32 所示, 当激活 Use Count 后会弹出一个 Group 管理列表, 表中将分别列出每一个物体在同一时间点的显示数量, 分别调整 Count 数值可控制每个物体的显示比例。

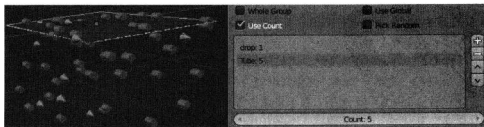


图 9-32 Use Count 面板对群组物体的管理

#### 9.1.5.5 Billboard (方块板) 模式

Billboard 模式下, 粒子将被替换为一种方块板物体, 常用于制作一些特殊的电视动画效果。由于每个 Billboard 都可以定义 UV 和光线追踪等效果, 因此它的渲染会消耗更多的内存和时间。如图 9-33 所示, 为 Billboard (方块板) 模式的参数面板。

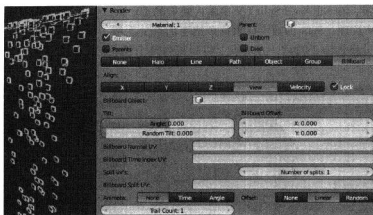


图 9-33 Billboard (方块板) 的参数面板

**Align:** 对齐方式, 定义了方块板从粒子源向外发射时的预对齐模式, X、Y 和 Z 分别将方块板对齐到对应的坐标轴上, 选择 View 时将取消对齐, 而 Velocity 则将按照速度法向来调整对齐方式。激活 Lock 锁定选项后, 方块板将在发射后的运动中始终对齐到所选的坐标模式。

**Tilt:** 倾斜度选项, Angle (角度) 值可控制方块板的旋转角度, 而 Random Tilt 可添加一个角度随机系数, 系数值越大, 方块板间的随机旋转角度差距也越大。

**Offset:** 用于调整方块板与粒子源中心的偏移距离。

**Billboard Normal UV/Time Index UV/Split UV:** 分别用于设置不同类型的 UV 贴图效果。以 Split UV 为例来讲解 UV 的添加流程。首先建立一个 UV 层, 进入网格属性, 单击 UV Texture 中的新建加号为粒子发射源添加一个 UV 纹理层, 如图 9-34 所示。



图 9-34 为源物体添加 UV 层

接着为物体材质添加一个纹理, 选择一个图片作为纹理贴图, 并在 Mapping (映射) 选项中将映射目标设置为刚才新建的 UV 层, 如图 9-35 所示。

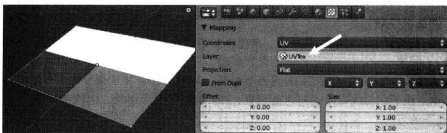


图 9-35 为材质添加 UV 纹理层

转到粒子面板，在 Billboard Split UV 中输入映射的 UV 源，也就是新建的 UV 层。Split 的原理是均匀平分 UV 纹理，Split 的数字代表每个边上的细分次数。由于纹理是一个  $2 \times 2$  的格子图案，所以如果细分设置为 2 的话，方块板上将只映射整个纹理的  $1/4$  图案。如图 9-36 所示，为粒子的设置面板。

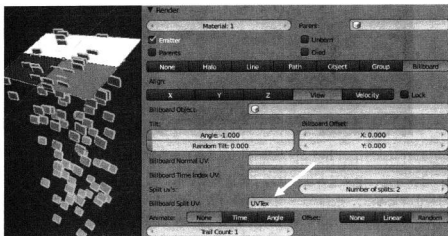


图 9-36 在粒子面板中添加 UV 数据层

单击快捷键 Alt + A 运行仿真，选择其中一帧进行渲染，效果如图 9-37 左图所示，可以看到每个方块板映射了 UV 的  $1/4$  图案。如果不使用 Split 选项的话，可以将数字调为 1 来取消细分操作，效果如图 9-37 右图所示，这里每一个方块板都继承了全部的 UV 纹理细节。



图 9-37 Split 的渲染结果

**Animate:** 动画选项能提供一个可选的辅助参数，来控制 UV 在动画中的变化效果。

**Offset:** 提供了两种方式来调整纹理的偏移量。

### 9.1.6 速率控制选项

可以通过改变粒子的初始状态参数，来控制粒子的速度与发射方向等运动属性，如图 9-38 所示，为粒子的 Velocity（速率）控制的参数面板。

**Emitter Geometry:** 粒子的初始速率参数设置，其中 Normal 用于调节粒子的法向初速度。

**Tangent:** 粒子在切线方向的初速度设置，Rot 数值用于为切线方向增加一个旋转系数，以控制切线速率的方向。

**Emitter Object:** 发射体选项，X、Y 和 Z 分别用于添加对应坐标轴上的加速度。

**Other:** 这里还可使用其他方式来调整粒子速度，如 Object 可使用发射体本身的速度来为粒

子赋予初速度，系数值越大。物体的速度对粒子影响越大。同时调整 Random 可添加一定的随机系数。

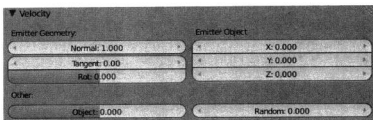


图 9-38 粒子 Velocity（速率）控制的参数面板

### 9.1.7 旋转控制选项

Rotation（旋转）参数可为粒子设置运动中的旋转属性。如图 9-39 所示，为 Rotation（旋转）控制的参数面板。



图 9-39 Rotation（旋转）控制的参数面板

**Initial Rotation:** 用于初始化粒子的旋转状态，单击旁边的小箭头弹出下拉菜单，可以选择一种约束角速度的模式。None 将默认使用全局的 X 坐标轴为约束方向，Normal 为粒子发射体的表面法向方向，而 Velocity 则为粒子的初始速度矢量方向。

**Phase:** 相位选项能控制旋转的相位属性，适当调大 Random 参数可增加一定的随机效果。

**Dynamic:** 这里的旋转约束设置控制着粒子的全程运动，如果激活 Dynamic（动态化）选项，粒子将由系统和环境引擎来自由模拟旋转的效果，初始化的设置将只能调整粒子的初始状态。

**Angular Velocity:** 定义了粒子角速度的转动模式以及速率，其中 Spin 为自旋模式，Random 则使用随机模式。转速参数用于定义角速度的大小，其值越大，粒子的自旋转越快。

### 9.1.8 子粒子系统选项

Children（子粒子）是一种特殊的无仿真效果粒子，它是在父粒子的运算基础上重新分配的次级粒子。对于一个粒子系统来说，如果粒子的数量较多，计算过程将花费相当多的资源和时间，来模拟碰撞和扰动等效果。可以使用较少的父粒子来执行模拟计算，然后让新添加的子粒子直接继承父粒子的物理计算结果，这样就可以在保持较好仿真效果的同时，又能节省大量的计算时间。

Children（子粒子）的控制方式和父粒子略有不同，如图 9-40 所示，为 Children（子粒子）系统选项的参数面板。

**Display:** 用于设置场景中粒子数量的可视化比例。

**Render:** 控制粒子实际渲染数占总数的比例。

**Radius:** 半径定义了子粒子围绕父粒子的分布半径范围，如图 9-41 左图所示，为 Radius 为 0 的粒子效果，当增大半径值后粒子的分布范围将扩大，如图 9-41 右图所示。

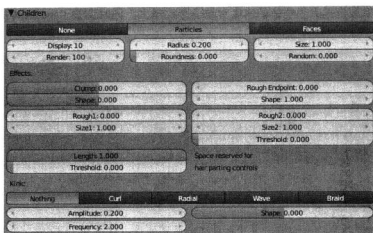


图 9-40 Children (子粒子) 系统选项的参数面板

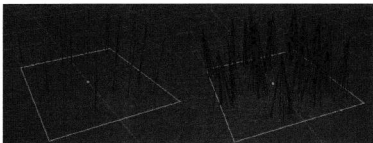


图 9-41 子粒子不同分布半径的仿真效果

**Roundness:** 设置子粒子围绕父粒子的顶端分布形状, 当值为 0 时, 粒子的顶端分布将为平面效果, 如图 9-42 左图所示, 当值为 1 时, 子粒子在末端的分布将是一个圆端的效果, 如图 9-42 右图所示。

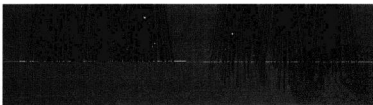


图 9-42 不同 Roundness 子粒子的分布仿真效果

**Effects:** 效果面板可控制子粒子的外形效果, Clump (粘结) 选项用于调整粒子的聚合位置, 当值为 -1 的时候粒子将在粒子源的位置产生聚合, 值为 1 的时候粒子则在末端产生聚合。如图 9-43 所示, 从左至右分别为 Clump 值为 -1、0 和 1 的效果。



图 9-43 不同 Clump 子粒子的粘连仿真效果



**Shape:** 外形选项, 可以使用曲线来控制子粒子的外形, 当值接近 0.99 时, 子粒子以反转的抛物线外形收敛, 而当值为 -0.99 时, 收敛方式将呈指数型曲线。如图 9-44 左图所示, 为 Shape 值是 0.99 时的粒子外形效果, 右图则为 -0.99 的粒子效果。



图 9-44 不同 Shape 粒子约束的仿真外形效果

**Rough Endpoint:** 末端粗糙化处理, 它的效果有点类似于在较低 Clump 值的基础上, 添加了一定的随机系数, 数字越大末端的随机扩散效果越明显, 如图 9-45 左图所示。Shape 则是在末端处理的基础上添加了外形控制, 当值小于 1 时, 外形将趋近于抛物线, 值较大时外形则呈双曲线模式, 如图 9-45 右图所示。



图 9-45 不同 Rough (粗糙化) 处理的仿真效果

**Rough1/Rough2:** 两个选项能提供更多自定义的粗糙化效果, 前者基于位移的偏移量来调整粒子路径, 后者则为粒子添加随机的向量属性, 其中 Threshold (阈值) 参数用于控制受影响的粒子数量, 值越小, 被粗糙化的粒子数量越少。如图 9-46 左图所示, 为使用 Rough1 的粗糙化效果, 右图则为 Rough2 的粗糙化效果, 其 Threshold 参数都设置为 0.5。



图 9-46 Rough1 和 Rough2 的粗糙化仿真效果

**Kink:** 绞结功能可以将子粒子按照一定的方式拧在一起, 聚拢在父粒子的周围, 常用于编织毛发等效果。Curl (盘绕) 形、Radial (放射) 形、Wave (波浪) 形和 Braid (发辫) 形模式分别提供了不同类型的绞结模式。如图 9-47 所示, 分别为 Curl、Radial、Wave 和 Braid 的绞结效果。

**Amplitude:** 幅度值设置, 这对旋转没有任何影响, 只影响子粒子的偏移幅度效果。

**Frequency:** 频度参数用于控制子粒子的偏移量, 值越高, 粒子的旋转越快。

**Shape:** 形体参数, 用于调整旋转的初始偏移位置。

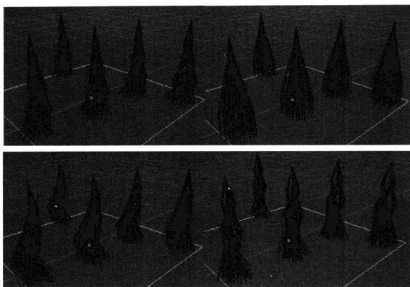


图 9-47 不同 Kink (绞结) 模式的仿真效果

### 9.1.9 毛发选项

当把粒子的 Type (类型) 切换至 Hair (毛发) 时, 粒子将不再以颗粒的方式向外发射, 而是使用静态的路径显示方式来模拟毛发的效果。毛发的长度由粒子的发射速度来决定, 初速度越大, 毛发越长, 而毛发的数量则由粒子的数量来决定。

粒子毛发还可以和其他物理模拟配合工作, 例如 Soft (软体) 仿真。但是要注意的是: 物理模拟本身就会占用较多的系统内存资源, 当配合粒子运算时, 每根毛发都将被计为一个软体物体, 所以这时应尽量减少参与计算的主粒子数, 使用添加子粒子的方式来丰富毛发细节。

#### 9.1.9.1 Vertex Groups (顶点组) 控制

可以使用自定义顶点组来控制不同区域的毛发密度。如图 9-48 所示, 为粒子的 Vertex Groups (顶点组) 控制的参数面板。

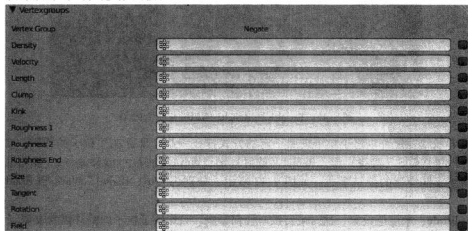


图 9-48 Vertex Groups (顶点组) 控制的参数面板

顶点组控制是一种十分灵活的粒子调节工具，粒子的每一项属性都可由一个或多个网格顶点组来执行约束。如图 9-49 左图所示，在 Weight Paint（权重喷涂）模式下，为球体绘制一个自定义的权重分布图，并将该权重顶点组应用到 Density（分布密度）通道中，这样粒子的实际毛发效果将如图 9-49 右图所示，其中权重值越高的区域毛发将越密集，反之则越稀疏。

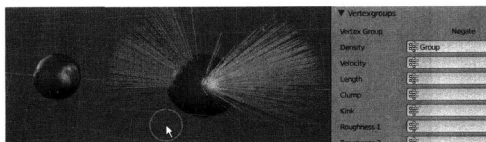


图 9-49 顶点组对粒子的属性约束效果

### 9.1.9.2 Hair Dynamics（毛发动力学）选项

毛发的参数面板中会新增一个 Hair Dynamics（毛发动力学）选项，如图 9-50 所示，为其参数面板。

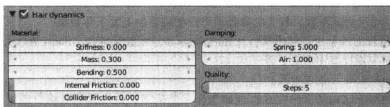


图 9-50 Hair Dynamics（毛发动力学）选项的参数面板

当开启毛发动力学后，系统将使用物理模拟来完成毛发之间的碰撞计算，单击 Alt + A 即可执行仿真运算。

**Stiffness:** 这里的 Material（材质）属性用于调节毛发的物理属性，其中 Stiffness 定义了毛发的初始运算位置。如图 9-51 左图所示，为 Stiffness 值为 0 时的效果，这时粒子的计算位置将从根部开始。如图 9-51 右图所示，为增大 Stiffness 至 0.5 时，毛发从总长的一半距离处开始计算动力学的效果。

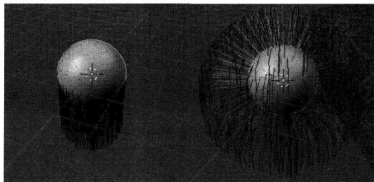


图 9-51 不同 Stiffness 初始运算位置的仿真效果

**Mass:** 粒子质量, 用于添加粒子的重量感。

**Bending:** 曲度值, 用于控制毛发的褶皱系数, 值越大, 代表粒子的褶皱幅度越大, 也就是向外弯曲的幅度值越大。如图 9-52 左图所示, 为 Bending 为 0 的粒子效果, 当将值增加到 5 时, 效果如图 9-52 右图所示。

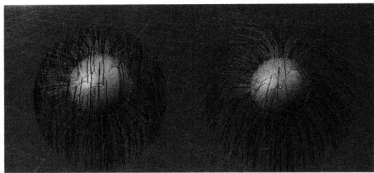


图 9-52 不同 Bending (曲度值) 的仿真效果

**Internal Friction:** 内摩擦系数, 用于设置毛发之间的摩擦系数。

**Collider Friction:** 碰撞摩擦, 用于控制毛发与环境互动时的摩擦效果。

**Damping:** 阻尼衰减系数, 较大的参数值可产生较慢的毛发下落效果。

**Quality:** 仿真的质量由每帧中执行计算的 Steps 步进系数来控制, Steps 值越大, 仿真的质量越高, 但是相对来说计算速度会更慢。

#### 9.1.9.3 Particle Mode (梳理模式)

有些毛发效果无法由仿真来完成, 例如制作一条辫子, 这时就需要将自动模拟切换为手动控制, 借助 Particle Mode (梳理模式), 来完成这种自定义结构的毛发编辑。在梳理模式下, 每一根毛发都将以 Segment (分段细分) 的方式进行均匀细分, 每个细分点都是一个控制点, 使用类似笔刷的批处理工具, 来调整这些点的位置以达到梳理的效果。

如图 9-53 左图所示, 添加一个默认的 Monkey (大圣) 物体, 当添加粒子毛发系统后, 为其定义一个区域顶点组, 并使用这个顶点组来约束粒子密度, 效果如图 9-53 右图所示。

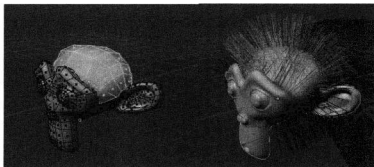


图 9-53 使用顶点组控制大圣的毛发生长区域

这时如果直接开启 Hair Dynamics, 单击 Alt + A 执行仿真运算, 毛发将按照默认的物理环境设置下垂, 如图 9-54 左图所示。这样的效果并不理想, 为了给大圣理一个漂亮的发型, 单击视图窗口中的模式选择, 进入 Particle Mode (梳理模式), 如图 9-54 右图所示。



图 9-54 进入 Particle Mode (梳理模式)

单击 T 调出梳理工具列表, 如图 9-55 所示。其中 Size (尺寸) 选项用于调整笔刷的大小, 而 Strength (强度) 则控制笔刷的作用力。默认的笔刷对毛发的修改效果影响是正向的, 如果在操作的同时按住 Shift 可执行反向化效果。



图 9-55 梳理工具列表

梳理模式的操作十分简单, 只需要选择对应的笔刷工具, 就可以在毛发上随意执行调整与修改操作了。

**Comb:** 合并工具可将毛发往一个区域集中, 它对毛发的选择效果类似于一种衰减操作。

**Smooth:** 平滑工具, 能将毛发上的结点做均匀平滑处理。

**Add:** 添加工具, 可以在笔刷经过的位置上添加毛发。

**Length:** 延展工具, 能将当前选择的毛发拉长或者缩短。

**Puff:** 膨胀工具, 可以将毛发向笔触位置集中。

**Cut:** 削减工具, 使用笔刷的边缘来修剪毛发长度。

**Weight:** 权重工具, 这常与软体模拟配合使用, 权重值越低的软体影响效果越明显。

梳理的毛发有三种显示方式, 同时每种显示方式可使用不同的选择模式, 可以在标题栏上的小图标选项上做模式切换。如图 9-56 左图所示, 为 Path (路径) 模式, 这种模式下的毛发节点不可见, 所有的操作影响效果将被均匀应用在毛发上。如图 9-56 中图所示, 为 Point (结点) 模式, 这时毛发上的所有结点均可见, 对毛发的操作将替换为对所有点的操作。如图 9-56 右图所示, 为 Tip (顶点) 模式, 这种模式下系统将仅显示毛发的末端结点, 笔刷工具通过改变末端部分的位置, 来反向影响整根毛发的形态。



图 9-56 三种毛发选择模式

当选择使用 Point 和 Tip 模式时, 对毛发结点的编辑类似于对网格物体上点的操作。可以使用 RMB 来选择单个结点, 也可配合 Shift 来选择多结点。单击快捷键 A 可选择全部结点, 而快捷键 B 则可开启框选操作。当选择一根毛发的任意位置结点时, 单击快捷键 L 还可自动选择相连的整根毛发。

使用快捷键 W 可调出特殊选择模式, 如图 9-57 所示, 为 Specials (特殊) 选择模式的菜单。

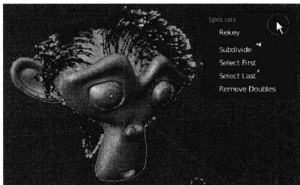


图 9-57 Specials (特殊) 选择菜单

**Rekey:** 删除所有的线段结点。

**Subdivide:** 可对所选择的毛发做细分操作。

**Select First/Last:** 用于分别选择毛发的顶端和根部结点。

**Remove Doubles:** 可移除重复的结点, 常用于清理多余结点的操作。

## 9.2 烟雾仿真系统

Smoke (烟雾) 是一种特殊的粒子效果, Blender 为其单独提供了一套 Smoke (烟雾) 仿真工具。如图 9-58 所示, 为使用 Smoke 工具制作的火焰效果。



图 9-58 使用 Smoke 模拟的火焰效果

### 9.2.1 烟雾域选项

要完成 Smoke 仿真，首先需要在场景中新建一个 Domain（域），因为每个烟雾系统都需要一个 Domain（域）来定义烟雾的扩散范围，所有产生的烟雾都无法穿透这个 Domain（烟雾域）。

#### 9.2.1.1 定义 Domain（域）

以 Cube 物体为例，首先在 Physics（物理）面板中找到 Smoke（烟雾）的选项，如图 9-59 所示，为切换至 Smoke（烟雾）效果的 Domain（域）参数面板。

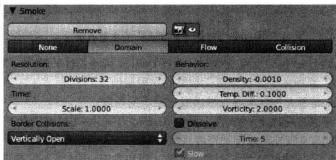


图 9-59 Domain（域）的参数面板

**Resolution:** 烟雾分辨率，以 Divisions（细分等级）来控制烟雾的精度。

**Time:** 控制仿真的速度比例。

**Border Collisions:** 域的碰撞边界选项，用于设置烟雾与域碰撞时的运算对象。

**Behavior:** 烟雾的物理属性，其中 Density（密度系数）值越大，烟雾的上升速度越快；而 Temp Diff（温差系数）可模拟烟雾受温度的影响程度，值越高，烟雾的上升速度越快；Vorticity 则为烟雾的躁动系数。

**Dissolve:** 激活后可以控制烟雾将在多长时间后逐渐消失，Time 为烟雾持续的时间设置。

#### 9.2.1.2 Smoke High Resolution（高分辨率烟雾）选项

域的设置决定了烟雾的主体，而激活 Smoke High Resolution（高分辨率烟雾）选项，可以为烟雾添加类似于粒子子的细节效果。如图 9-60 所示，为其参数选项面板。

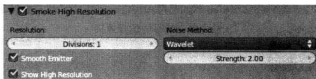


图 9-60 Smoke High Resolution（高分辨率烟雾）的参数选项

**Resolution:** 高分辨率烟雾的 Divisions（细分等级）选项。

**Smooth Emitter:** 平滑发射器，避免出现雾效堵塞在发射器表面上的现象。

**Show High Resolution:** 可开启高分辨率的显示功能。

**Noise Method:** 系统通过增加噪点来计算烟雾的细节，这里提供了两种方式，通常使用 FFT。

### 9.2.2 烟雾发射体选项

在 Cube 的中下方添加一个小 Cube 物体，并在 Physics（物理）面板中将其定义为 Smoke 物体，接着选择它的 Type（模式）为 Flow（烟雾流）。如图 9-61 所示，为 Flow（烟雾流）的参数面板。

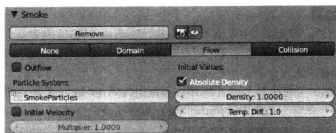


图 9-61 Flow (烟雾流) 的参数面板

**Outflow:** 默认的 Flow 模式可生成烟雾, 选择 Outflow 可将物体转换为一个烟雾出口。

**Particle System:** 添加粒子系统来强化烟雾的发射属性。

**Initial Velocity:** 是否允许烟雾效果继承发射粒子的速度。

**Initial Values:** 烟雾的初始行为属性, 其中 Absolute Density (绝对密度) 选项可使烟雾保持发射时的密度值, Density 则控制烟雾的密度系数, Temp Diff 为温差系数。

系统将自动为小 Cube 物体添加一个粒子系统, 可以在 Particle System 选项中, 对粒子属性做调整和修改。完成所有设置后, 选择 Domain 物体, 单击 Bake 或使用快捷键 Alt + A, 仿真效果如图 9-62 所示, 烟雾将从底部 Flow 物体处升起, 当接触到 Domain 的顶部时即停止下来。

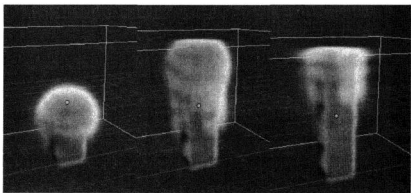


图 9-62 烟雾的仿真效果

### 9.2.3 碰撞与障碍选项

可以在系统中添加一些物体, 并将其定义为障碍物, 用于阻碍烟雾的运动。首先在 Inflow 物体的上方添加一个 Sphere (球体), 并单击物理面板, 将其定义为 Smoke (烟雾) 的 Collision (碰撞器), 如图 9-63 所示, 为 Collision (碰撞器) 的参数面板。



图 9-63 Smoke (烟雾) 的 Collision (碰撞器) 参数面板

调整好球体的尺寸和位置, 单击 Bake (烘焙), 物体对烟雾的阻碍效果如图 9-64 所示。可以看到, 烟雾在接触球体后, 将会受阻而向四周散开。



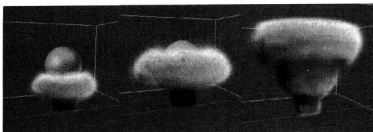


图 9-64 物体与烟雾的碰撞效果

### 9.2.4 烟雾渲染选项

烟雾的渲染方式较为特殊，需要特定的 Shade（着色器）设置。首先选择烟雾的 Domain，为其添加一个 Volume 材质，并且将 Density 设置为 0，Density Scale（尺寸）设置为 2，Scattering 增大至 2，同时将 Step Size 缩小为 0.010，具体参数设置如图 9-65 所示。

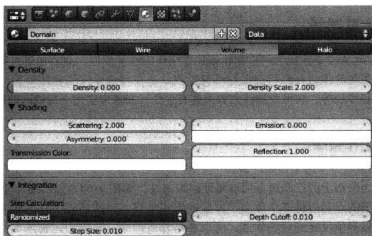


图 9-65 Volume Shade（体着色器）的参数设置面板

转至 Texture 面板，将贴图属性修改为 Voxel Data，在 Domain Object（域对象）中输入烟雾域的名称，这个例子中为 Cube，最后在 Influence 中开启 Density 控制，具体设置如图 9-66 所示。

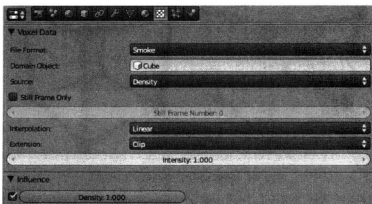


图 9-66 Texture（纹理）的参数设置面板

如图 9-67 左图所示，为在场景中重新模拟的烟雾效果。接着为场景添加灯光并单击渲染，烟雾最后的材质效果如图 9-67 右图所示。

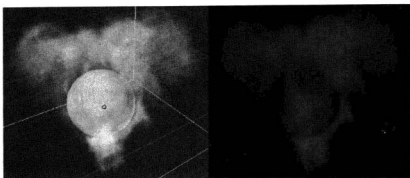


图 9-67 Smoke 的渲染效果

## 9.3 软体仿真系统

Soft（软体）动力学是系统为网格物体添加非刚性的物理属性，然后通过对物体与周围环境的碰撞模拟，来表现软体的弯曲和挤压等形变过程。常在动画中应用软体模拟来制作一些特殊物体，例如果冻和橡胶等，它可令物体的表面拥有弹性，在重力作用下与周围的环境交互中产生碰撞和反弹效果。

### 9.3.1 软体工作原理

Blender 的软体动力学的工作原理是将一个物体细分为多个单元体，将每个单元体转换为一个弹性网格，并为网格上的点和边分别添加可伸缩属性，如图 9-68 左图所示，为一个弹性网格的结构图。网格上的点和边将产生类似于弹簧的阻尼效应，使网格体可在内部实现弹性的挤压效果。由于真实世界中没有绝对的软体，所以在与外界的交互碰撞中，物体本身存在的刚性属性，将使网格上点和边在挤压和拉升时的形变幅度呈曲线型衰减。如图 9-68 右图所示，可以近似地对边和点添加一种液压模拟，来实现软体在伸缩至一定幅度时产生的缓冲效应，这种调整可以让仿真更接近于物理的真实。

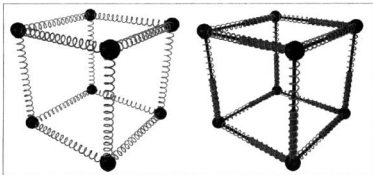


图 9-68 软体动力学的网格结构模型

启动软体模式和启动粒子仿真时一样，都可以使用快捷键 Alt + A 来快速执行缓存仿真计算。一旦达到了满意的仿真预览，即可单击 Cache 缓存选项中的 Bake（烘焙），将模拟数据保存为硬

盘数据。烘焙仿真数据可以节约大量的重复模拟运算时间,同时实现物理模拟的快速回放。如图 9-69 所示,为 Cache (烘焙缓存) 的参数面板。

Start 和 End 定义了仿真的时间段,单击 Bake 按钮可直接烘焙场景中的物理运算。Cache Step (烘焙步长) 可调整烘焙的间隔帧数,激活 Disk Cache 后可将烘焙结果按照 Cache Name (命名) 格式保存在硬盘上。

由于软体的模拟是一种自行交互的计算,所以在设置好物体的物理与环境的属性后,就可以直接运行软体计算了。烘焙中影响软体模拟的因素主要包括重力、风力和斥场等场景外力,软体与环境的 Collision (碰撞),以及 Self Collision (物体自身内部的作用力) 等。如图 9-70 所示,为 Matt Ebb 制作的果冻动画。

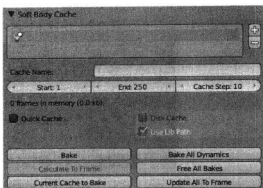


图 9-69 Cache (烘焙缓存) 的参数面板

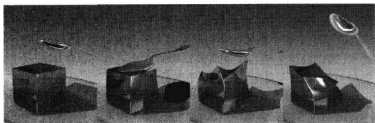


图 9-70 Matt Ebb—Soft Body Gelatin

### 9.3.2 软体的外部作用力

系统在计算外力对软体的作用效果时,默认采用了牛顿物理定律。当软体在不受外力作用时,物体将保持运动的初始状态,当有外力作用于软体上时,网格上结点的加速度与物体的质量和所受外力大小相关,同时结点与结点之间还存在作用力和反作用力效果。根据软体动力学中的网格细分原理,软体支持所有由结点组成的物体,包括网格物体和曲线物体,同时物体的网格数越多软体的仿真精确度也越高。如图 9-71 左图所示,新建一个 Cube 物体,并做 2 次细分处理,然后再添加一个细分修改器,最后的效果如图 9-71 右图所示。

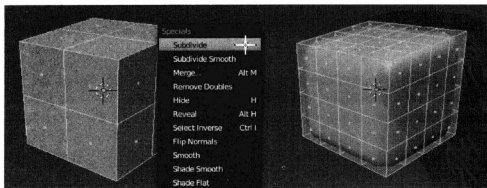


图 9-71 创建一个软体本体

### 9.3.2.1 Soft Body 软体选项

回到物体模式，在 Physics（物理）选项面板中，单击 Soft Body（软体）属性中的 Add（添加）按钮，系统将自动把 Cube 转换为一个软体物体。如图 9-72 所示，为 Soft Body（软体的）参数面板。

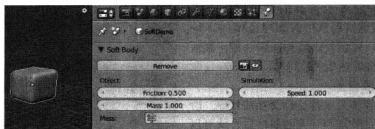


图 9-72 为物体添加 Soft（软体）属性

**Friction:** Object（物体）属性中定义了软体的一些基本物理特性，其中 Friction（摩擦）属性值用于设置作用在网格所有结点上的通用摩擦力，数值越大，软体在接触面上的移动越困难，当设置为 50 时，物体将完全无法滑动。

**Mass:** 质量值用于定义物体的结点质量系数，物体的质量越重它的运动也越缓慢，同时它的运动惯性也越大。当赋予了一个物体软体属性时，系统将默认把所有的点都计入质量计算，计算公式为物体质量 = (结点数量 × 结点质量系数) / 1000。这里也可以使用顶点组来定义区域性的软体效果，只需要在 Mass 后填入自定义的顶点组名称即可。

**Speed:** Simulation（仿真）选项下的 Speed（速度）增量系数，用于控制软体系统的内部时间，值越大，物体的运动速度相对全局中的其他物体会越快。

### 9.3.2.2 Soft Body Goal（目标体）选项

Soft Body Goal（目标体）选项，可以将软体定位在某一个特定的位置上，或者根据父子关系将软体吸附在另一个物体上。在激活 Goal 选项时，软体将只承受外力对其本体属性的影响，而这些外力主要来自于被吸附物体的 F-Curve 动画和骨骼动画所驱动的运动加速度，或者父系物体运动时产生的加速度效应。如果关闭 Goal（目标体），软体将成为自由运动的仿真体，例如在场景中执行自由落体运动。如图 9-73 所示，为 Soft Body Goal（目标体）的参数面板。

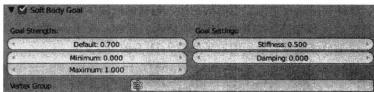


图 9-73 Soft Body Goal（目标体）的参数面板

**Goal Strengths:** 这里定义了软体本身的弹性属性，Default（默认值）实际上是软体在静态时受重力的影响系数，系数值越小，物体的弹性效果越明显。

**Minimum/Maximum:** 最小值和最大值分别用于控制物体在运动过程中弹性系数的波动范围，如图 9-74 所示，左图为 Default 值为 0.2 时的软体效果，中图为增大值至 0.7 的软体效果，如果此时将 Minimum 和 Maximum 分别调节至 0.5 和 1，那么拖动软体时其自身弹性效果将如图 9-74 右图所示。

**Goal Settings:** 目标体选项中可设置其 Stiffness（刚度）系数和 Damping（阻力）系数，分别用于调整物体的弹性值和其与空气的阻力值。Stiffness 数值越大时软体的弹性伸缩幅度越小，

Damping 值越大时, 软体的运动越缓慢。

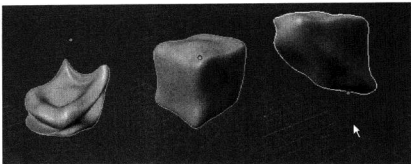


图 9-74 不同弹性系数值下的软体效果

**Vertex Group:** 可以将物体的部分网格固定在目标体上, 只需要使用 Vertex Group 来添加特定的顶点组即可。

#### 9.3.2.3 Soft Body Field Weights (软体场权重) 选项

当同一个外力作用在多个软体共存的环境中时, 例如地球的重力或者固定的风向, 可以通过设置软体的场权重值, 使不同物体可以对同一个外力产生各异的影响效果。如图 9-75 所示, 为 Soft Body Field Weights (软体场权重) 的参数面板。

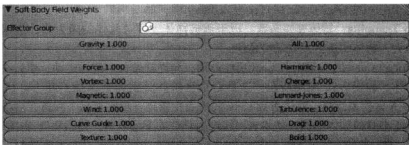


图 9-75 Soft Body Field Weights (软体场权重) 的参数面板

可以在 Effector Group 中输入指定的自定义外力群组, 默认时将计算整个场景中的所有外力。以 Gravity (重力) 系数为例, 当值为 1 时, 当前软体将受到重力值 100% 比例数值的影响, 如果降低重力比例系数, 软体受到的重力影响效果将相应的降低, 如果设置为 0, 那么该物体将区别于场景中的其他物体, 产生完全不受重力影响的效果。其他的场权重系数原理类似, 只需要根据不同的环境变量做适当的调整即可。

### 9.3.3 软体的碰撞计算

#### 9.3.3.1 Collision (外部碰撞体) 选项

除了外力或重力等正向作用力对物体产生的软体效果, 碰撞也是软体与环境产生交互的一种方式。当软体和物体产生碰撞时, 接触的碰撞表面将对软体产生反作用力, 用于阻止并抵消软体的正向运动。要讨论软体碰撞, 需要首先制作一个碰撞体, 选择一个物体, 在物理面板中的 Collision (碰撞) 选项中单击 Add 按钮, 将其转换为带碰撞属性的物体。如图 9-76 所示, 为 Collision (外部碰撞体) 选项的参数面板。

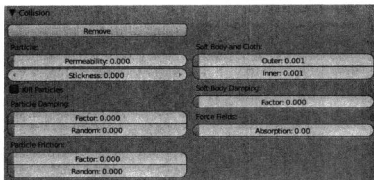


图 9-76 Collision (外部碰撞体) 选项的参数面板

**Soft Body and Cloth:** 软体与布料的碰撞体选项, 其中 Outer 和 Inner 分别用于设置碰撞体表面外层和物体内层对软体的缓冲范围。当软体进入带碰撞属性的物体 Outer (外侧) 范围内时, 软体将受到缓冲效果而产生减速运动, 反作用力在软体上将产生相应的形变效果。Inner 为从物体表面向内的障碍填充范围, 用于设置物体内部的缓冲范围, 如果软体在碰撞时速度足够快或者质量足够大, 那么它可能穿过 Outer 层直接嵌入 Inner 层, 如图 9-77 所示。

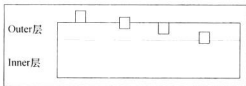


图 9-77 Outer 和 Inner 碰撞层范围原理图

**Soft Body Damping:** 软体摩擦系数, 用于调整物体表面对软体的阻力值, 值越大碰撞时的摩擦阻力也越大, 物体的速度也降低得越快。

将前面建立的 Cube 的 Soft Body Goal 属性取消, 允许其做自由落体运动, 然后在其正下方添加一些台阶物体, 如图 9-78 左图所示。选择每个台阶并分别为其添加 Collision (碰撞) 属性, 然后单击 Alt + A 做模拟运算。可以看到 Cube 将在自由落体一段距离后, 在落在台阶后开始向下滚动, 同时由于本身的软体属性, 软体表面将产生弹性的变形效果, 如图 9-78 右图所示。

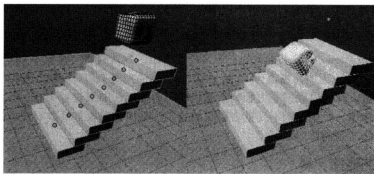


图 9-78 软体的碰撞互动效果

### 9.3.3.2 Soft Body Collision (软体的自碰撞) 选项

除了调节软体与外界物体的碰撞参数, 还需要调整软体本身在碰撞时的自碰撞特性。如图 9-79 所示, 为 Soft Body Collision (软体的自碰撞) 参数面板。

软体的内部结点主要由 Ball (结点球) 组成, 自碰撞计算实际上就是计算 Ball (结点球) 之间的空间压缩效应, 所以这里的自碰撞选项也重在设置 Ball (结点球) 的属性。系统提供了 5 种

Collision Type (结点球类型), 常选择 Average 让系统根据网格的边长来计算结点球的平均尺寸。

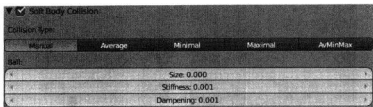


图 9-79 Soft Body Collision (软体的自碰撞) 参数面板

**Size:** Ball (结点球) 的属性设置中, Size 定义了结点球的大小, 值越大结点球间的间距越短, 碰撞后的伸缩幅度也越窄, 因此软体看上去会更紧绷, 但是计算量也将更大。

**Stiffness:** 刚性系数, 值越大, 结点的弹性越弱, 当 Size (尺寸) 为 0 时将默认关闭刚性系数。

**Dampening:** 阻力系数, 值越大, 结点在碰撞时速度的衰减越快。

如图 9-80 所示, 将 Size 设置为 1, Stiffness 设置为 10, 而 Dampening 设置为 0.4。单击模拟, 对比图 9-78 右图的效果可以发现, Cube 物体在与阶梯产生外部碰撞后, 将在内部结点之间产生自身碰撞, 结点间的距离在压缩后将反向增大, 由于调高了 Size 和 Stiffness 系数, 物体不会产生大比例的弹性形变, 同时由于设置的 Dampening (阻力) 系数较大, 软体的滚动将逐渐缓慢, 并且在阶梯的半途位置停了下来。

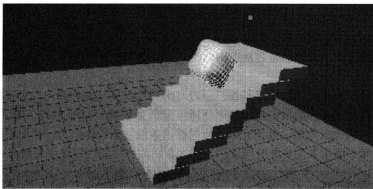


图 9-80 添加自碰撞的软体模拟效果

### 9.3.4 软体的内部作用力

前面在讲软体动力学的原理时, 将结点与结点之间的边抽象为一种可伸缩的弹簧, 这样的网格边能有效地模拟结点间的挤压效果, 但是这样的结构并不稳定, 一旦结点受到较大外力作用时, 受力碰撞的结点很可能会翻过与它相邻的结点而产生破面现象, 甚至导致软体完全塌陷, 如图 9-81 左图所示。那么解决的方法是除了增加自碰撞计算, 还需要优化结点间的存储结构, 通过添加更多的对角边来强化网格结构, 例如使用插值积分的算法来固定结点所有可能的有效运动方向, 如图 9-81 右图所示。只有通过结合内部作用力和外部作用力的影响效果, 才能使软体的运动趋近于稳定。

#### 9.3.4.1 Soft Body Edges (软体边) 选项

如图 9-82 所示, 为软体的 Soft Body Edges (软体边) 选项的参数面板。

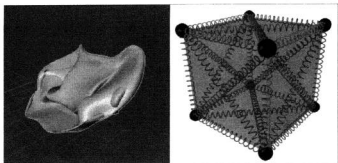


图 9-81 通过调整内部作用力来强化软体的网格结构模型

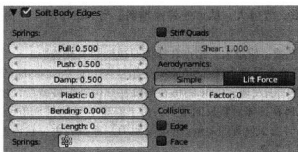


图 9-82 Edges (边) 选项的参数面板

**Pull:** Springs 压力属性定义了软体边的近弹簧性能, 其中 Pull 为边的弹性拉升系数, 值越低, 边的弹性越弱, 值越高软体将越接近刚性物体。一般来说, 对于橡胶物体取 0.5, 对于餐巾或皮革等物体可取 0.999。

**Push:** 边的挤压系数, 用于定义其抵抗压力时的弹性硬度, 值越高时软体的刚性力度越强, 对充气物体的仿真可取较高的值, 而织物等物体可取低系数值。如图 9-83 左图所示, 为 Pull 和 Push 分别为 0.1 时的仿真效果, 软体此时由于弹性值太低导致物体因为过软而坍塌掉。如图 9-83 右图所示, 为将两个值增大至 0.9 时的效果, 相比左图软体的刚性效果明显增强了。

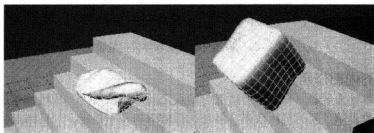


图 9-83 Pull 和 Push 两种弹性系数对软体的影响效果

**Damp:** 摩擦系数, 用于控制软体边与其他物体之间的摩擦力。

**Plastic:** 塑性系数, 值越大软体在碰撞后越难恢复原来的弹性性能。如图 9-84 左图所示, 为 Plastic 值为 0 时软体的落地效果, 可以看到其结构能很快恢复到最初下落时的形状。而如图 9-84 右图所示, 为将 Plastic 增大到 90 时的效果, 这时的软体由于自身的恢复性能大大降低, 在经历一系列的碰撞后几乎无法再恢复到其原始的形态了。



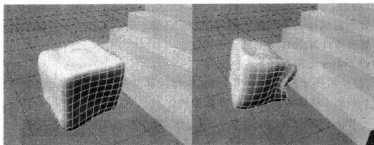


图 9-84 Plastic（塑性）系数对软体碰撞后的形态恢复影响效果

**Bending:** 刚度混合系数，用于控制结点与其周围结点的刚度混合值，系数值越大的软体，其结点间的弹性刚度也越大，物体在碰撞后反弹时将会显得更硬。

**Length:** 可重定义软体边的初始弹簧长度，当值为 0 时将关闭此功能，低于 100 时软体边将对原长度做等比例数值的缩放计算。如图 9-85 左图所示，为 Length 为 80 的效果，这时软体在碰撞后将被缩小到原尺寸的 80%，而右图则为 120 时的软体效果，此时的物体在碰撞后被放大至原尺寸的 120%。

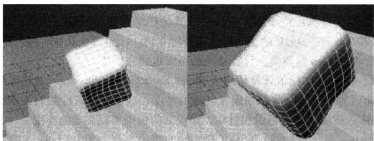


图 9-85 Length 对软体尺寸的影响效果

**Stiff Quads:** 象限硬度，激活后系统将为每个面的对角线分配硬度系数，用于防止软体的面产生过度塌陷。其中 Shear 值定义了对角线的刚度系数，值越大物体的对角刚性值也越大。如图 9-86 左图所示，为开启 Stiff Quads 的效果，而右图是关闭后的仿真结果，可以发现软体上的面在碰撞一次后就完全塌陷掉了。

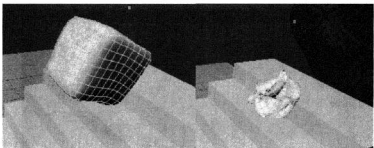


图 9-86 Stiff Quads 选项可防止软体出现面塌陷的错误

**Aerodynamics:** 为软体模拟添加空气动力学效应，使得软体在运动时受到空气或者垂直于软体边的风力等外界阻力影响，一般来说，将这里的系数设置为 0.3。而 Simple 和 Lift Force 则用于确定边界的受力情况，选择后者时的计算效果将更精确。

**Collision:** 开启前, 软体的运动和碰撞只针对结点, 当激活 Edge (碰撞边) 和 Face (碰撞面) 后, 系统将分别对软体的边和面积做碰撞效应计算。开启 Collision 选项后能弥补大量碰撞中的计算错误, 但是模拟计算的时间也会显著增加。

#### 9.3.4.2 Soft Body Solver (软体解算器) 选项

Solver (解算器) 可用于精确控制 Blender 执行运算时的算法性能和误差矫正, 如图9-87所示, 为 Soft Body Solver (软体解算器) 的参数面板。

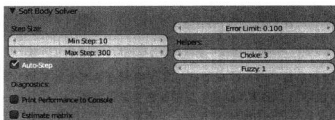


图 9-87 Soft Body Solver (软体解算器) 的参数面板

**Step Size:** 步进值选项规定了系统对软体的采样计算量, 其中 Min Step 定义了每帧的最少采样计算次数, 而 Max Step 则定义了每一帧的最大采样计算次数, 其中选择 Auto Step 将根据运动速率来做自适应的采样计算。由于软体系统的计算采用的是插值算法, 所以对于设置每秒为 25 帧的回放率, 如果 Min Step 为 10 的话, 那么系统将在每一帧也就是 0.004 秒内至少做 10 次插值采样运算, 因此, Max Step 值越高, 计算结果的精度也越高。

**Error Limit:** Blender 的插值计算使用了 Runge Kutta Correct Physics (龙格库塔) 法则, 系统将使用 Error Limit (容错极限) 参数来定义计算的精度, 这也是碰撞检测的关键设置之一。较低的数值可提高运算的精度、减少错误和抖动, 而较高的值则可提高运算速度。

**Choke:** Helpers (碰撞辅助器) 定义了碰撞时的相关参数, 其中 Choke (抑制) 系数用于控制软体结点或边在进入碰撞网格范围, 也就是接触 Outer 时, 缓冲速度的计算系数。

**Fuzzy:** 模糊系数用于修订碰撞时计算的容差范围, 值越大处理速度越快, 但会变得不稳定, 常结合 Error Limit 容错系数来平衡计算的稳定性和运算的时间。

**Diagnostics:** 诊断系统, 开启 Print Performance to Console 后可将运算情况打印输出到控制台, 用于查错和监控。而 Estimate Matrix 则会在软体的重心处添加一个坐标轴, 方便在计算中实时观察软体的碰撞运动, 如图 9-88 所示。

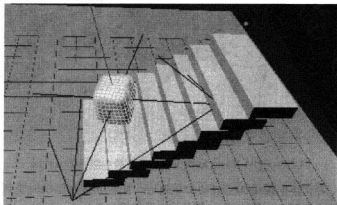


图 9-88 Estimate Matrix (坐标显示) 模式

## 9.4 布料仿真系统

Cloth（布料）模拟是 CG 制作中的常用物理仿真之一，例如制作衣服、旗帜和纸张等物体。Blender 的布料模拟系统建立在通用的空气动力学仿真系统基础上，使得布料不仅可以和与其接触的物体产生碰撞运动效果，还能受重力、风力等外力影响产生互动效果，例如披风和窗帘的飘动等。

可以将布料本身看做是一种特殊的软体，但是区别于软体模拟的弹簧网格结构计算模型，布料模拟器采用的是网格互动的方式来模拟外形的变化。如图 9-89 所示，为 Daniel Genrich 制作的 Mancandy 布料演示。

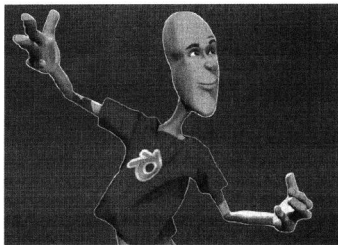


图 9-89 Daniel Genrich—Mancandy shows off his Blender T-shirt

### 9.4.1 布料初始化

由于布料的计算以网格为交互计算的基本模型，所以网格的面数越多，运算的精度也就越高，但是相应的运算时间和资源消耗也会更大。首先在场景中添加一个大圣物体，如图 9-90 所示，同时在它的上方添加一个 Plane 物体。接着单击 Tab 进入编辑模式，将 Plane 做 3 次细分操作，并添加一个细分修改器。

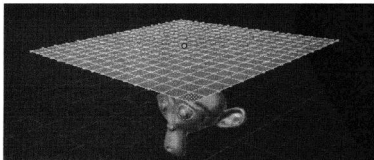


图 9-90 建立布料运算场景

回到物体模式，在选择 Plane 物体的情况下，进入其 Physics（物理）面板，单击 Cloth（布料）选项上的 Add 按钮，即可将 Plane 物体定义为一个布料物体，如图 9-91 所示。

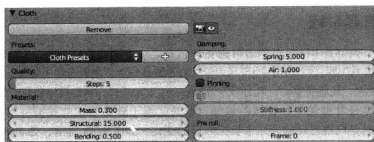


图 9-91 Cloth (布料) 的参数面板

**Presets:** 选用预设的布料属性，单击旁边的十字箭头也可新建布料属性。这里提供的预设包括 Cotton (棉)、Denim (牛仔)、Leather (皮革)、Rubber (橡胶) 和 Silk (丝绸) 等类型。

**Quality:** 品质系数，Steps 值越高，布料的运算结果越精确。

**Material:** 布料的物理属性，包括 Mass (质量)、Structural (硬度) 系数和 Bending (褶皱) 系数。

**Damping:** 阻力系数，包括 Spring (弹性阻力) 和 Air (空气阻力) 系数，它们的值越大，布料在外力作用下的速度衰减也越快。

**Pinning:** 可以使用顶点组将布料的一部分固定挂起来，其中 Stiffness (强度) 系数用于调整固定点的牢固程度，如果系数值较小，那么布料被悬挂的部分越容易被外力移动。

**Pre roll:** 设置预模拟的帧数。

在完成布料设置后，为大圣添加一个 Collision (碰撞) 属性，这时如果单击 Alt + A 激活仿真，布料的运算效果如图 9-92 所示。可以看到布料物体从高处落下时，在受到空气阻力后产生了软体形变，当接触到大圣时，布料由于重力作用和低摩擦力将顺势继续下滑。

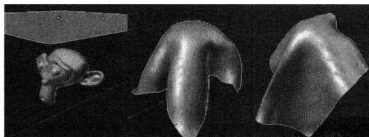


图 9-92 布料垂直下落的仿真效果

## 9.4.2 布料碰撞选项

为了确保布料的正确模拟，需要根据不同的场景需求，调节布料与外界的碰撞条件。同时布料和碰撞物体必须被存放在同一个图层中，并开启 Cloth Collision (布料碰撞) 功能。如图 9-93 所示，为 Cloth Collision (布料碰撞) 的参数面板。



图 9-93 Cloth Collision (布料碰撞) 的参数面板

**Quality:** 品质系数, 值越大, 结果越精确, 同时布料产生撕裂与穿透的几率也越小。

**Distance:** 碰撞时物体与布料间的最小碰撞作用范围, 当物体与布料间的距离小于这个数值时, 系统就会认为它们之间产生了碰撞效果, 布料开始执行碰撞后的偏转运算。

**Friction:** 摩擦力系数, 用于控制布料的光滑程度, 值越小, 布料在物体上滑动的阻力越小, 运动也越快。

**Self Collision:** 开启自碰撞, 用于避免布料自碰撞时产生的穿透错误。其中 Quality (自碰撞品质系数) 用于调整碰撞时的解算次数, Distance 则定义了布料间的最小碰撞作用范围。如图 9-94 左图所示, 为未开启自碰撞时的布料效果, 可以看到布料内部产生了相互穿透的现象。开启自碰撞后的效果如图 9-94 右图所示, 布料间避免了错误的计算结构, 但是仿真的时间也相应增加了。

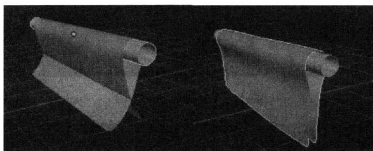


图 9-94 Self Collision (自碰撞) 计算效果

**Collision Group:** 使用顶点组来自定义布料上产生碰撞效果的网格范围。

### 9.4.3 布料硬度控制选项

Cloth Stiffness Scaling (布料硬度控制) 是布料的高级选项, 当开启硬度选项后, 系统将根据顶点组的权重值来自适应地调整布料的硬度效果。如图 9-95 所示, 为 Cloth Stiffness Scaling (布料硬度控制) 的参数面板。



图 9-95 Cloth Stiffness Scaling (布料硬度控制) 的参数面板

**Structural Stiffness:** 输入开启硬度控制的网格顶点组, 在下面的选项栏中输入顶点组名称, 并为其定义结构硬度值即可。

**Bending Stiffness:** 同理, 为所选择的顶点组设置最大的弯曲硬度值。

### 9.4.4 外力场选项

Blender 提供了多种物理外力, 用于配合各种仿真运算的环境互动, 在 3D 视图中单击 Shift + A 即可在新建菜单中选择 Force Field (外力场), 如图 9-96 所示, 为二级菜单中的外力场类型。

以 Wind (风力) 为例, 如图 9-97 所示, 为 Wind (风力) 的参数面板。

**Strength:** 风力的强度值, 值越大, 风力越强。

**Flow:** 模拟的空气流速强度值。

**Noise:** 为风力添加一定的噪性系数。

**Seed:** 噪性的随机性系数。

**Effect point:** 分别控制受风力作用时, 对象的 Location (位移) 和 Rotation (旋转) 属性变化。

**Collision:** 风力的碰撞属性, 用于控制风的作用力是否能在碰撞后被吸收。

**Falloff:** 用于选取风力的衰减模式。

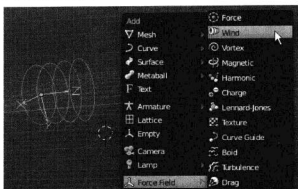


图 9-96 Force Field (外力场)

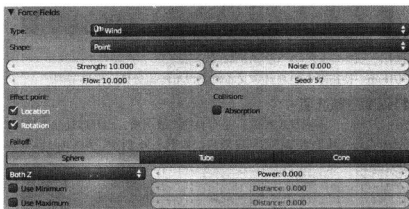


图 9-97 Wind (风力) 的参数面板

将风力对准布料, 单击 Alt + A, 效果如图 9-98 所示, 布料随风力的作用漂浮了起来。

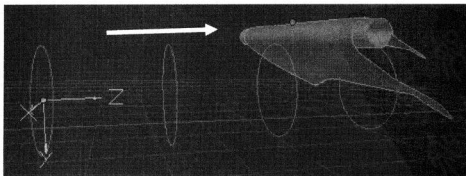


图 9-98 布料与风力的交互作用效果

### 9.4.5 烘焙选项

当设置好布料属性后，就可以将仿真结果烘焙保存起来了。如图 9-99 所示，为布料的 Bake（烘焙）的参数面板。

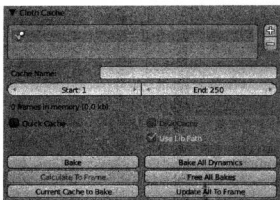


图 9-99 Bake（烘焙）的参数面板

和软体的烘焙选项类似，这里的 Start 和 End 分别定义了仿真的时间段，单击 Bake 按钮可直接烘焙场景中的物理运算结果至本地数据文件，激活 Disk Cache 后即可将烘焙结果按照 Cache Name 的命名格式保存在硬盘上了。

## 9.5 流体仿真系统

Blender 的流体动力学基于牛顿运动学算法，用于简单模拟液体在固体界壁间的流动效果。自然界中的流体是由大量不断作热运动，而且无固定平衡位置的分子构成，它的基本特征就是没有稳定的形状，并且具有流动性和可压缩性。在流体的形状被改变时，流体各层之间将产生一定的运动阻力，也就是流体的粘滞性。当流体的可压缩性和粘滞性值很小时，就可被近似地看作是理想的流体，Blender 的流体模拟都是基于这种理想流体来实现的。如图 9-100 所示，为 Mike Pan 制作的《Strawberry》。



图 9-100 Mike Pan—Strawberry

## 9.5.1 流体原理

Blender 的流体系统采用了 El'Beem 开源流体模拟库, El'Beem 是基于 LBM (Lattice-Boltzmann method, 格子玻尔兹曼算法) 的一个高效物理流体模拟库, 关于 El'Beem 库的更多信息, 请到它的官方网站<sup>①</sup>查询。

LBM 是一种 CFD (Class Fluid Dynamics, 计算流体动力学), 它是由 LGA (Lattice Gas Automata, 格子气自动机方法) 发展而来的。LBM 主要应用非连续介质的思想来研究宏观的物理现象, 它使用网格技术来模拟被离散的流体分子, 在每一个等距的单元栅格中都存储一个离散的速度值, 来等价运算当前位置的流体分子, 同时栅格的分布函数也表示了流体中粒子沿着其运动方向的速度矢量。可以使用有 19 个矢量方向的 D3Q19 模型来表示这种栅格结构, 如图 9-101 所示。

LBM 不直接处理质点的运动过程, 而是以充满运动流体质点的流场空间为对象, 计算各时刻质点在流场中的变化规律。这种对单个流体质点运动过程置之不理, 而只专注于流场中各空间点的运算方法, 称为流场法。

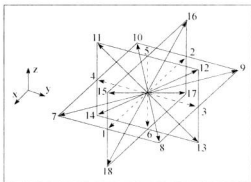


图 9-101 流体的 D3Q19 模型

## 9.5.2 流体域设置

### 9.5.2.1 Domain (域) 选项

和 Smoke (烟雾) 模拟一样, 每个流体系统也都需要一个 Domain (域), 用于定义流体的运算空间。所有的流体都将被约束在 Domain (域) 内, 无法流出流域空间之外。一套流体系统中仅允许存在一个流域, 但是可以允许使用多个流体源。一般来说可以使用最简单的 Cube 来制作 Domain (域), 因为流体系统中的流体模拟和域的形状并无关系, 它只是用于约定一个计算范围。如图 9-102 所示, 新建一个 Cube 物体, 在物理面板中单击 Fluid (流体) 选项下的 Add 按钮, 即可添加一个流体系统, 接着将其 Type 属性设置为一个 Domain (流体域)。

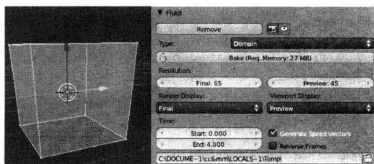


图 9-102 新建 Domain (流体域) 系统

**Bake:** 用于激活流体烘焙, 并将运算结果保存至本地指定的文件目录。对于精度较高的烘焙计算, 需要准备足够的磁盘空间, 建议至少空余 2G 以上。

① 网站地址: <http://elbeem.sourceforge.net>。



**Resolution:** 流体精度，也就是流体的粒度属性。精度值越低，流体的表面越粗糙，反之将越精细。精度值决定了流体的层次细节，同时也决定了运算的时间和磁盘使用量，高精度的流体运算需要消耗大量的内存资源，当精度设置为200时，可能需要高达495 MB的内存来参与运算。其中 Final 用于设置最后的渲染精度，而 Preview 则用于设置预览时的精度显示比例。如图9-103左图所示，为 Final 精度为20的仿真效果，而右图则是 Final 精度为80的仿真效果，它们都基于100%的 Preview 精度查看值。

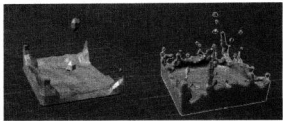


图9-103 不同 Resolution（流体精度）值的效果

**Display:** 显示模式，Render Display 选项可设置渲染时的流体显示模式，而 Viewpoint Display 可选择场景中的流体显示模式。这里总共有三种模式，其中 Final 为 Final Resolution（最终精度）下的流体显示效果，Preview 则显示 Preview Resolution（预览精度）下的流体显示效果，选择 Geometry 将仅显示最原始的网格效果。

**Time:** 烘焙时间，分别设置烘焙的开始时间和结束时间，这里默认的时间设置单位为秒，所以如果 Start 为0，End 为4，那么针对25FPS的场景来说，烘焙数量就为100帧。

**Generate Speed Vectors:** 为流体建立动态模糊缓存。

**Reverse Frames:** 是否反转烘焙的流体帧。

#### 9.5.2.2 Domain World（域环境）选项

如图9-104所示，为 Domain World（域环境）选项的参数面板。

**Gravity:** 重力设置，这里可根据需求和环境设定来配置影响流体的重力属性，一般都保持默认的9.8数值。如果关闭场景选项中的全局重力设置，则可以在这里设置重力的自定义值。

**Viscosity Presets:** 粘度预设，流体的粘度决定了它的移动速率和它与物体间的摩擦力。由于现实中流体粘度还受温度的影响，例如热巧克力的粘度就因温度而异，而 Blender 在这里只提供了标准的常温模拟，所以可使用不同的粘度值来计算不同温度下的流体效果。预设中可选的粘度系数包括 Water（水）、Honey（蜂蜜）和 Oil（油），当然也可以使用 Manual 来手动设置自定义粘度。如图9-105所示，为 Manual（自定义）粘度输入的参数面板。

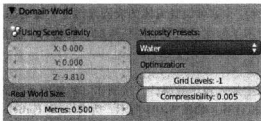


图9-104 Domain World（域环境）选项的参数面板

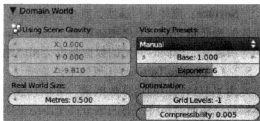


图9-105 Manual（自定义）粘度输入的参数面板

在输入自定义流体粘度时应注意，真实世界中的粘度计算通常采用 Dynamic Viscosity（动力粘度，也就是 Poiseuille 泊肃叶流动定律）来描述流体层间相对运动时产生的内摩擦阻力，以 Centi Poise（厘泊）或 Pascal-seconds（帕斯卡秒）来定量测定。而 Blender 系统仿真使用的是工程计算中常用的 Kinematic Viscosity（运动粘度），来表示重力作用下流动时的内摩擦力度量，其值为动力粘度与密度的比值，计量单位为 Centi Poise（厘泊）。

自定义面板中，Base（基数）用于设置动力粘度基数，Exponent 则输入粘度指数。例如常

温下的水动力粘度为 1.002 cP 厘泊,也就是  $1.002 \times 10^{-6}$  Pa·S 帕斯卡秒,所以 Base 中应输入 1.002, Exponent 中则输入 6。对于融化的钢铁或玻璃来说,由于其非常粘稠,粘度大概为  $1.000 \times 10^6$  cP,所以这时的 Base 中应输入 1.000,而将 Exponent 设置为 0 即可。注意:系统的流体仿真对象不适合材料不流动的非流动体,所以即使设置很高的粘度值也不会让物体呈现刚体行为。如表 9-1 所示,为一些常用流体的粘度系数表。

表 9-1 常用流体的粘度系数表

流体	动力粘度 (cP)	运动粘度 ( $\text{m}^2 \cdot \text{s}^{-1}$ )	流体	动力粘度 (cP)	运动粘度 ( $\text{m}^2 \cdot \text{s}^{-1}$ )
水 (20℃)	1.002	$1.002 \times 10^{-6}$	巧克力糖浆	30 000	0.003
油	500	$5.0 \times 10^{-5}$	番茄酱	100 000	0.1
蜂蜜 (20℃)	10 000	0.002	融化玻璃	$1.0 \times 10^{15}$	1.0

**Real World Size:** 环境比例,用于适当地设置仿真环境和真实环境的标尺比例,这里以每个 Blender 单位等价于现实中的尺寸来控制流体的质量等物理属性。

**Optimization:** 流体的优化设置,其中 Grid Levels (网格层级) 用于设置模拟期间系统将使用多少个栅格层,当值为 -1 时,系统将做自动选择。适当地调节 Compressibility (压缩比) 可处理高精度流体的运算错误,但是会相应增加运算时间。

#### 9.5.2.3 Domain Boundary (域边界) 选项

除此之外,还需要对 Domain 中流体与物体 Boundary (边界) 间作用力等属性做相应的设置。如图 9-106 所示,为 Domain Boundary (域边界) 的参数面板。



图 9-106 Domain Boundary (域边界) 的参数面板

**Slip Type:** 黏性流类型,用于设置流体与障碍物之间的黏性程度,这里可供选择的类型包括 No Slip (黏性)、Free Slip (无黏性) 和 Partial Slip (部分黏性),其中 Partial 的效果将介于两者之间。使用 Amount 系数则用于调整黏性的力度。

**Surface:** 流体表面属性,其中 Smoothing (平滑) 参数可设置流体的表面光滑度,值越高,流体的表面越光滑,反之则越粗糙。Subdivisions 则用于创建高精度的网格表面,值越大,细分的等级越高,但同时网格的数据量也会相应地增大。

#### 9.5.2.4 Domain Particles (域粒子) 选项

为了节约模拟的运算数据量,可以在流体模拟中加入粒子效果,直接制作一些极细小的流体颗粒,从而避免将所有的细节都交由纯流体来模拟计算。如图 9-107 所示,为 Domain Particles (域粒子) 的参数面板。

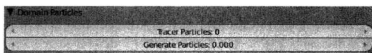


图 9-107 Domain Particles (域粒子) 的参数面板

**Tracer Particles:** 跟踪粒子数,用于在流体模拟初期添加跟踪流体运动的粒子数量。

**Generate Particles:** 粒子的生成数量控制,值越大,粒子的数量越多。为了能体现较好的效

果, 建议将粒子的数量至少设置在 2 级以上。如图 9-108 左图所示, 为没有粒子效果的流体, 当加入粒子后, 流体效果如右图所示, 可以看到画面中增加了不少的水滴颗粒效果。

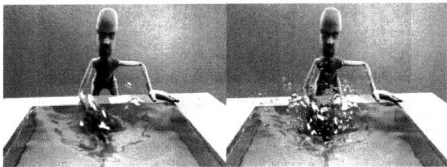


图 9-108 带粒子效果的流体模拟

### 9.5.3 流体设置

#### 9.5.3.1 Fluid (定容积流体) 源

当建立好了 Domain (域), 就可以在域范围内添加流体源, 用于向域中注入流体。如图 9-109 所示, 在 Cube 中添加一个球体, 并单击 Fluid (流体) 面板上的 Add (添加) 按钮, 将其转换为一个 Fluid (定容积流体) 源。

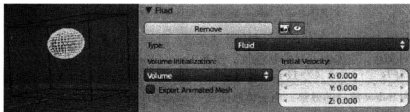


图 9-109 定容积流体属性的参数面板

Fluid 定容积流体将根据网格尺寸来计算注入的流体体积, 这是一种固定容积的流体模式, 并且流体在域内不会增加或者流失。

**Volume Initialization:** 流体体积的初始状态, 用于定义流体的初始容量计算方式。其中 Volume 将初始化网格内部空间为流体的体积, 但这仅针对封闭的网格物体。Shell 则只计算网格表面, 这种模式下的液体体积根据表面积来决定, 所以可用于非封闭的物体。Both 则将前两者结合起来计算流体容积, 但仍仅针对封闭的网格物体。如图 9-110 所示, 从左至右分别为 Volume、Shell 和 Both 三种流体的初始状态。



图 9-110 Volume Initialization (体积的初始状态)

**Export Animated Mesh:** 如果网格拥有动画属性, 那么激活这个选项可开启交互模拟。

**Initial Velocity:** 流体的初始速度, 用于计算流体在模拟开始时的初始速度。为 Fluid 指定一

个 Z 正向的初速度，然后选择 Domain 并单击 Bake（烘焙），流体的下落效果如图 9-111 所示。

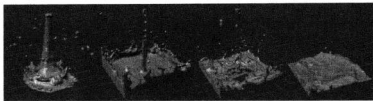


图 9-111 具有初速度的 Fluid（流体）模拟效果

#### 9.5.3.2 Inflow（注入式流体）源

域中的流体除了可以使用这种定容积注入的方式，还可以使用 Inflow（注入式流体）源，持续不断地向域中注入流体。如图 9-112 所示，在域中添加一个较小的 Cube，设定其 Type 为 Inflow（注入式流体）源。



图 9-112 Inflow（注入式流体源）的参数面板

**Volume Initialization:** 流体的输入初始状态选项，同 Fluid 模式。

**Inflow Velocity:** 流体输入时的初速度，激活 Local Coordinates 将为流体入口使用本地坐标系，这可以让流体入口在发生位移和旋转动画时，流体能跟随做同步的移动和旋转动作。如图 9-113 所示，为 Inflow 流体的效果，当流体碰撞到 Domain 的边界时，将沿着边壁流向域的底部。

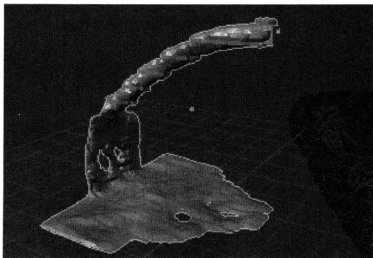


图 9-113 Inflow（注入式流体）

#### 9.5.3.3 Outflow（流体出口）

同理，可以设置 Outflow（流体出口），在域的底部添加一个 Cube，并将其属性设置为 Out-

flow（流体出口）。如图 9-114 所示，为 Outflow（流体出口）的参数面板。

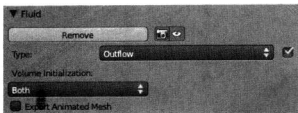


图 9-114 Outflow（流体出口）的参数面板

Outflow 的参数很简单，只定义流体出口的初始状态模式，和前面的其他类型流体源类似。使用定量 Fluid 将域灌满，当开启模拟后可以看到，随着时间的推移，域中的流体将逐渐减少，并从 Outflow（流体出口）处流失掉，如图 9-115 所示。

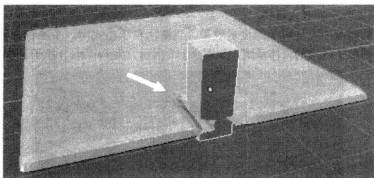


图 9-115 流体 Outflow（流体出口）

## 9.5.4 交互控制

### 9.5.4.1 障碍模式选项

可以在域中添加一些障碍物，用于阻拦流体的运动。紧接着前面的例子，在 Cube 中添加一个 Plane 物体，单击 Tab 进入编辑模式，调整其倾斜度。接着进入其物理面板的 Fluid 选项，单击 Add 将其转换为流体物体，并将属性设置为 Obstacle（障碍）属性。如图 9-116 所示，为 Obstacle（障碍）属性的参数面板。

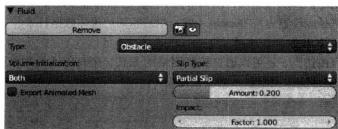


图 9-116 Obstacle（障碍）属性的参数面板

**Volume Initialization:** 障碍体体积计算初始状态，和流体的配置原理一样。

**Slip Type:** 黏性类型，和流体的黏性属性类似，No Slip 黏性最大，Free Slip 表面最光滑，黏性最小。如图 9-117 所示，左图为 No Slip 的流体效果，中图为自定义黏性系数为 0.2 的效果，而右图

则是 Free Slip 的效果。可以看到, 物体表面由于十分的光滑, 几乎无法附着任何流体。

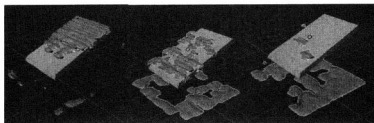


图 9-117 不同 Slip (黏性) 属性下的流体碰撞效果

**Impact:** Factor 为受动因素, 用于表示障碍物移动时流体体积的流失修正值, 如果障碍物没有移动, 这个修正值就不起作用。如果障碍物移动并且与流体碰撞, 设定为负值时将从流域中减少流体的体积, 正值则效果相反, 一般来说取值范围应为  $-3 \sim 3$ 。

#### 9.5.4.2 Particle (粒子) 模式选项

可以在流体的碰撞中强化粒子的显示效果, 当激活粒子模式时, 系统将为物体在当前位置创建一个用于模拟粒子效果的流体系统。如图 9-118 所示, 为 Particle (粒子) 模式的参数面板。

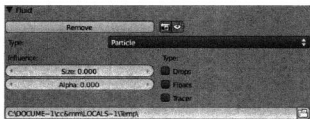


图 9-118 Particle (粒子) 模式的参数面板

**Influence:** 影响值, 包括 Size (尺寸) 和 Alpha (透明度), 当 Size 为 0 时, 所有粒子的尺寸大小均相同, 系数值越大, 粒子的尺寸变动范围越大。当 Alpha 值为 0 时, 所有的粒子透明度都相同, 而且, 值越大时, 尺寸越大的粒子将获得越低的 Alpha 值。

**Type:** 粒子类型包含 3 种, Drops (水滴) 型可用于模拟流体表面张力较低时, 飞溅产生的大量小水滴; Floats (浮动) 型则用于模拟表面较大张力的流体, 例如海水, 因为张力越大流体就会显得越重, 所以飞溅出的水滴会以更快的速度聚合起来并回落至表面; Tracer (追踪) 类型能令大量的水滴跟随流体表面运动, 例如水面上悬浮起来的薄雾效果。

**Path:** 这里的路径应设置为与 Domain (域) 烘焙相同的流体数据目录。

#### 9.5.4.3 Control (流体控制器) 选项

除了让流体做自由仿真, 还可以通过控制网格的方式, 来自定义地控制流体的运动。如图 9-119 所示, 为 Control (流体控制器) 的参数面板。

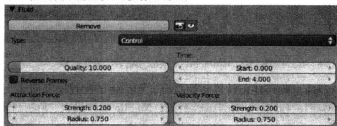


图 9-119 Control (流体控制器) 的参数面板

**Quality:** 品质系数, 数值设置越高, 流体的可控粒子数量越多, 细节的层次越丰富, 同时计算的时间也越长。

**Reverse Frames:** 是否开启流体的反向运动计算。

**Time:** 用于设定网格物体对流体的控制时间范围, 当仿真时间超过这里的 End 时间时, 流体将脱离控制做 Domain (域) 环境中设置的自由重力运动。

**Attraction Force:** 指定网格物体控制流体的作用力大小, 其正值能将流体吸附起来, 负值则将流体从表面反向排斥。其中 Strength (强度) 值用于调节作用力的大小, 而 Radius (半径) 则设置网格表面到流体的法线距离。

**Velocity Force:** 物体运动对流体产生的作用力设置, 当物体在控制流体运动时, 自身移动的速度也会对流体产生一个加速度作用力。和 Attraction Force 中一样, 这里采用 Strength (强度) 值和 Radius (半径) 来调节作用力的控制力度和作用范围。

如图 9-120 所示, 为 Nils Thuerey 在 Siggraph 2007 上演示的流体控制。它使用一个网格来控制流体的运动, 并最终释放控制让其回落至容器域中。



图 9-120 Nils Thuerey—Magic Fluid Control

### 9.5.5 流体仿真优化

使用计算机来模拟流体运动是一件十分困难的事情, 因为无论采用何种算法, 计算机都始终无法模拟出流体的真实物理运动。在这里对流体的模拟其实也只是一种近似仿真, 并不是真正的物理还原, 是以欺骗人眼的方式来获得流体运动视觉效果的手段。流体仿真需要消耗大量的时间和系统资源, 所以, 越能深入地理解流体动力学的原理, 越能高效地得到令人满意的效果。如图 9-121 所示, 为使用 Yafaray 渲染的流体仿真。

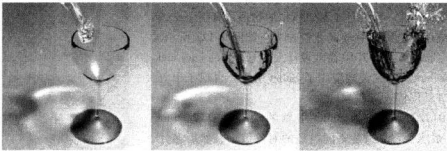


图 9-121 使用 Yafaray 渲染的流体仿真

经过实验可以发现, 在 Blender 的流体设置中, Domain (流体域) 的空间越大, 仿真的时间越长; 黏性系数越低, 流体的速度越快, 对应的系统消耗也就越大。所以掌握流体的几个关键属性原理, 并且平衡它们之间的设置, 是获得最佳性能模拟结果的唯一方式。同时, 在仿真的初期应尽量使用较低的预览设置来获取近似结果, 而在最后的渲染输出时才应该提高各相关的属性参数。

如果要制作一个拥有大量流体仿真的场景，可以尝试将 Domain 分解，因为域的空间越大，使用的内存量也越大。可以尝试将每个 Domain 分别执行烘焙，然后将数据存放在本地磁盘，在渲染的时候统一读取即可。如果大量的数据造成渲染出错，也可以使用分层渲染，将不同 Domain（域）的流体渲染在不同的层中，最后使用结点工具来完成重叠与合成。

流体本身的运动速度越快，黏性系数越低，在碰撞中就会产生大量的插值数据，这时如果解算器设置不合理的话，就可能无法捕捉这些数据，从而产生大量的错误和误差。其实仿真中的流体并不需要使用真实世界中相同的速度值，所以可以尽量降低这些参数值，以免造成过于繁琐的计算。

在仿真的过程中，磁盘会自动生成大量的临时数据文件，而且它们不会被自动删除掉。所以在每一次仿真测试之前和结束之后，请手动清空这些数据文件，以免造成回放动画时出现数据的交叉误读错误。

## 9.6 刚体仿真系统

在外力作用下，与环境中的其他物体发生碰撞等交互时，不产生形状等物理改变的物体被称为 Rigid Body（刚体）。刚体是一种理想的物理模型，事实上，大自然中的每一种物体在受到外力作用下都会产生一定的形变，但在 3D 仿真中，可以忽略这些细微的变化。刚体仿真就是使用物理引擎，来模拟这种理想刚体效果的物理运动学。

物理引擎的主要任务是检测碰撞，解决碰撞和执行相关约束，为 3D 环境中的所有物体提供基于物理原理的行为更新。通常可根据应用领域将物理引擎划分为两大类，Real-Time（实时）物理引擎和 High-Precision（高精度）物理引擎。前者由于降低了精度并简化了运算，所以能够以合适的帧率实时地演算物理碰撞，常应用于游戏和 VR 虚拟现实等交互式程序中。后者则由于更复杂的算法和更高的精度要求，常需要十分强大的计算机和非常长的运算过程，通常被应用于科学计算。

### 9.6.1 Bullet 引擎简介

Blender 的物理模拟没有对刚体的模拟，要实现刚体的运动模拟，只能在 BGE（Blender Game Engine）游戏引擎下运行。Bullet 是集成于 BGE 系统中的一款开源物理引擎，它包含了多线程的 3D 碰撞检测和刚体动力学库，是一款专注于游戏模拟的物理计算引擎，现在也常被应用于电影特效等商业制作中。如图 9-122 所示，为 Blender Game Contest 2010 游戏设计大赛冠军作品 Lucy and the time machine，它完美地应用了 Bullet 引擎，再现了十分真实的物理动作特效。

如图 9-123 所示，为 Bullet 引擎的动态计算 Pipeline（管线）。系统会首先封装不同类型的对象数据，然后在计算重力等作用力时，按照前向动力学的顺序执行物理仿真计算，当完成交互检测后，再将物理效果按照模块输出整合起来，并在场景中实现行为更新。

有关 Bullet 的更多信息请登录它的官方网站<sup>①</sup>查询。

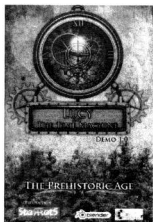


图9-122 Vitor Balbão & Bernardo F. Hasselmann—Lucy and the time machine

① 网站地址：<http://bulletphysics.org/>。



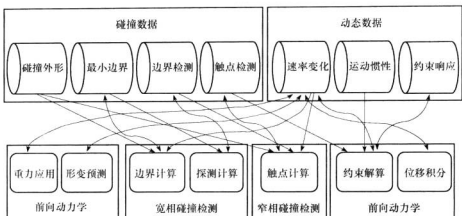


图 9-123 Bullet 的物理计算 Pipeline (管线)

### 9.6.2 环境配置

在执行刚体运算之前,首先需要切换至 BGE 环境,如图 9-124 所示。在环境配置中,单击下拉菜单选择 Blender Game (游戏) 模式。

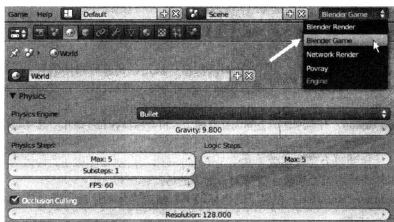


图 9-124 切换至 BGE 环境

接着切换至 World (世界) 面板中,这里新增了一个 Physics (物理) 选项,选择将要使用的 Bullet 物理引擎。

**Gravity:** 重力属性,用于设置环境中的全局重力值,一般保持默认的 9.80。

**Physics Steps:** 这里用于设置每一帧中的物理运算次数,Substeps (子步长) 的值越大,计算精度越高,FPS 帧率越高,视图的显示将越流畅。

**Logic Steps:** 每秒物理逻辑运算的最大次数。

**Occlusion Culling:** 定义了使用 DBVT 树运算时,每一帧上的闭塞缓冲分辨率,值越大,计算的速度越慢,但是精度也越高。

### 9.6.3 物理选项

对于任意的 Mesh (网格) 物体,均可将其转换为刚体。在选择物体的情况下,转入 Physics (物理) 面板,将物体类型设置为 Rigid Body 即可将物体转换为一个刚体。如图 9-125 所示,为 Rigid Body (刚体) 的物理参数面板。

**Actor:** 此项激活后,该物体才能被视为刚体,允许参与至环境中的物理计算。

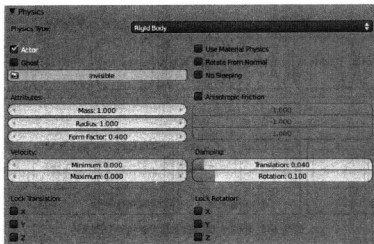


图 9-125 Rigid Body (刚体) 的物理参数面板

**Ghost:** 幽灵模式, 让物体拥有穿透其他物体的能力。

**Invisible:** 使物体在模拟中不可见。

**Use Material Physics:** 在材质选项中应用物理效果。

**Rotate From Normal:** 根据面的法线方向来控制物体的转动效果。

**No Sleeping:** 取消物体的自动休眠功能。

**Attributes:** 刚体的物理属性, 其中 Mass (质量) 决定了仿真时物体的重力, Radius (半径) 范围则定义了物体的有效碰撞边界范围, Form Factor 为碰撞的惯性因子, 数值值越小, 碰撞后的惯性作用力也将越小。

**Anisotropic Friction:** 开启刚体的各向异性摩擦力, 可令物体在每个轴上产生不同的碰撞摩擦力。

**Velocity:** 用于设置物体的运动初速度。

**Damping:** 物体的阻力衰减系数, 其中 Translation 用于控制物体移动时的阻力系数, 而 Rotation 则控制物体转动时的阻力系数, 值越大, 物体的运动越缓慢。

**Lock Translation/Rotation:** 在选中的坐标轴上, 锁定物体的运动和转动状态。

#### 9.6.4 仿真与记录

在场景中新建一个 Cube 物体, 然后挤压并排列成如图 9-126 所示的多米诺骨牌阵, 并将最后一个骨牌稍作倾斜。



图 9-126 搭建多米诺骨牌场景

将每一个 Cube 的物理属性都设置为 Rigid 刚体, 并添加碰撞属性。如图 9-127 所示, 为 Collision Bounds (碰撞边界) 的参数面板。



图 9-127 Collision Bounds (碰撞边界) 的参数面板

碰撞边界用于设置物体的碰撞边界类型，根据物体的形状来选择对应的边界类型即可。

完成设置后，单击场景面板中的 Start（运行）按钮，即可开启刚体的仿真运算，也可以使用快捷键 P。如图 9-128 所示，为运行后的效果，多米诺骨牌按照物理原理按顺序逐个倾倒。

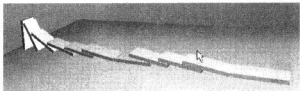


图 9-128 刚体的仿真效果

但是由于环境的不同，BGE 中的物体无法使用渲染器渲染，所以这时需要开启刚体的动画记录模式。如图 9-129 所示，单击标题栏中的 Game 菜单，激活 Record Animation（记录动画）选项。



图 9-129 Record Animation（记录动画）

这时再次单击 P，重新运行一次仿真，完成后再查看场景的 Graph Editor（曲线编辑器），可以发现，物体的运动被完整地记录成了 F-Curve 数据，如图 9-130 所示。这样当再次回到正常编辑模式时，只需要单击 Alt + A 就可以回放刚体仿真动画，并使用渲染器完成渲染了。

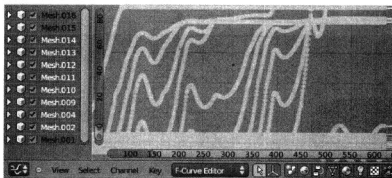


图 9-130 Graph Editor 中记录的刚体运动数据

## 第 10 章

## 后期合成

当完成渲染工作后，还需要使用一些后期工具，为渲染结果添加更多的细节处理，例如将各输出图层合并起来，强化部分通道的渲染细节，添加更多的自定义特效，弥补一些无法单纯由渲染来完成的效果图。Blender 提供了丰富的后期处理工具集，包括 Node（结点）工具和 Sequence（序列）编辑器。前者用于 Material（材质）、Texture（纹理）和 Render（渲染）的合成输出，而后者则常用于 Video（视频）的非线性编辑。Blender 的后期系统不仅适用于处理自身的渲染输出，也同样适用于导入本地图片或视频等文件。

## 10.1 结点系统

Nodes（结点系统）是一种过程流处理工具，每一个结点都是一个处理单元，需要为结点设置输入的数据源和相关处理的计算参数，以及最后的输出接口。结点系统中每一条从输入至输出的处理流水线又称为线程，一套完整的后期结点处理系统应该是由多线程的结点群组成的 Noodle Map（结点地图）。如图 10-1 所示，为《Creature Factory》<sup>①</sup>中一个场景使用的合成结点地图。

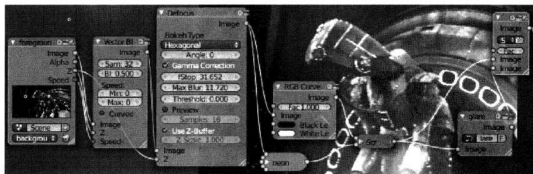


图 10-1 《Creature Factory》的合成结点地图

## 10.1.1 结点编辑器

## 10.1.1.1 激活结点合成模式

要激活后期的结点合成模式，需要在渲染前转至场景选项中 Post Processing（后期处理）面板，激活其中的 Compositing（合成）选项，如图 10-2 所示。

① 《Creature Factory》为 Blender 基金会出品的一套系列教学视频。



图 10-2 激活结点合成

当渲染完成后, 切换视图至 Node Editor (结点编辑器), 激活状态栏上的 Use Nodes 使用结点选项, 即可开始编辑结点地图或开启后期处理了。如图 10-3 所示, 为开启结点编辑器时的默认初始界面。



图 10-3 结点系统的初始界面

Node 标题栏上的三个小图标显示了当前编辑器的结点编辑模式, 分别为 Material (着色器)、Texture (纹理) 和 Render (渲染) 合成输出模式。Free Unused 按钮可清空未使用的结点, 以便释放更多的内存。Backdrop (背景图) 功能可开启渲染结果在背景中的显示, 激活后可使用快捷键 Alt + MMB 来移动背景预览图, 快捷键 V 和 Alt + V 可分别用于缩放背景图。这里需要注意的是, 背景预览图仅能显示当前激活 View 结点的输出画面。

#### 10.1.1.2 视图常用操作

对结点的操作和对物体的操作类似, 使用 RMB 来选择结点, 使用快捷键 G 移动结点, 删除结点可用快捷键 X。多选结点可使用快捷键 Shift + RMB, 并配合组合键 Ctrl + G 来群组结点。滚动 MMB 可放大或缩小结点地图, 而单击 MMB 可移动视图。如图 10-4 所示, 为结点系统中常用的操作快捷键菜单。



图 10-4 结点系统中常用的操作快捷键菜单

## 10.1.2 结点基础

### 10.1.2.1 Socket (接口)

每一个结点都是一个独立的模块体, 每个模块体分别拥有不同的处理计算功能。结点的输入和输出 Connectors (连接点) 又称为 Socket (接口), 可使用 LMB 从接口处引出连接线, 与相邻一级的结点建立连接关系。如果要断开连接, 可以使用快捷键 Ctrl + LMB 切断连接线即可。结点的输入接口都在结点边框的左边, 输出接口都在结点边框的右边, 所以结点的处理流程是按从左至右顺序执行的。结点的输入接口只支持单一的数据源输入, 但是输出接口支持多个分流输出。如图 10-5 所示, 为一对多的结点输出结构。

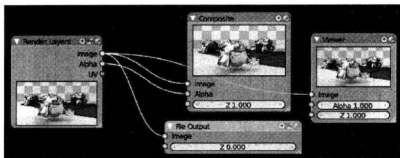






图 10-5 结点 Socket (接口) 的一对多连接方式

结点的 Socket (接口) 有三种不同的颜色, 分别代表不同的输出输入数据类型。如图 10-6 所示, 黄色的接口主要连接色彩数据, 灰色的接口主要传输数值类数据, 而蓝色的接口则主要用于传输矢量和法向等信息。通常结点间的连接应遵照同色互联的原则, 当然在某些时候交叉输入也能产生特殊的效果。

### 10.1.2.2 Collapse (折叠) 模式

单击结点左上角的箭头图标 , 可将该结点整个 Collapse (折叠) 起来。而右上方有 3 个小按钮, 可分别用来自定义结点的 Collapse (折叠) 部分。其中单击图标  可叠起没有输入或输出的接口, 单击  则用于叠起结点中的参数面板, 按钮  可将该结点的预览视图叠合起来。如图 10-7 所示, 为同一个结点在不同 Collapse (折叠) 方式下的视图效果。

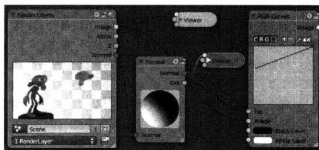


图 10-6 不同类型的结点 Socket (接口)



图 10-7 结点的 Collapse (折叠) 方式

### 10.1.2.3 Group（群组）功能

当结点地图中已布满了大量的结点，或者需要多次复用一部分结点所构成的处理功能时，可以利用结点的群组功能，精简结点编辑器中的结点地图，使编辑视图更整洁易于调用。如图 10-8 所示，为一个拥有大量结点的后期流程图，较多的交叉连线会让结点逻辑变得更繁杂凌乱，难以理解和管理。

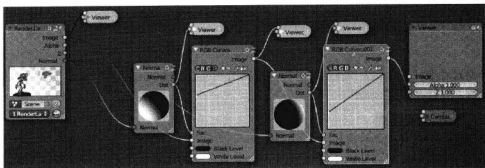


图 10-8 交叉连接的多结点地图

使用快捷键 B + LMB 选择中间的所有处理环节结点，然后单击快捷键 Ctrl + G，即可将这些结点群组为一个 Group（组）。如图 10-9 所示，为群组后的工作界面，可以看到工作区变得十分清爽。群组后的结点会将内部主要结点的参数接口显示在两侧，同时将主要参数的选项显示在群组主体上，只需要调整群组结点上的这些参数，即可调整整个处理流程。如果要重新设计流程内部，可以在使用 RMB 选择该群组的情况下单击快捷键 Tab，展开群组中的结点组合，执行编辑操作，如图 10-10 所示。

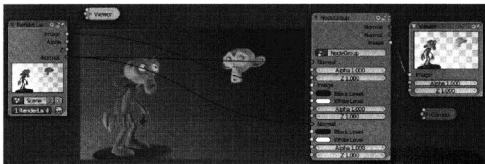


图 10-9 群组后的工作界面

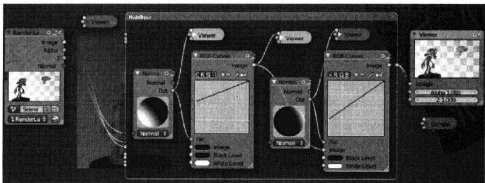


图 10-10 展开群组中的结点组合

## 10.2 着色器结点模式

着色器结点模式常用于制作一些特殊的 Material（结点材质）。要编辑结点类材质，首先需要在结点编辑器中将结点处理模式切换到材质模式，同时激活 Use Nodes 选项，如图 10-11 所示。

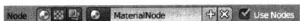


图 10-11 激活材质的结点模式

新建结点类材质的方法和新建普通类材质略有不同，可直接在状态栏中单击加号来新建结点类材质，并修改材质的名称。

### 10.2.1 输入类结点

#### 10.2.1.1 Material（标准材质）结点

在材质模式下，可以导入不同类型的数据作为合成的素材，最常用的是导入系统的默认 Material（材质）。如图 10-12 所示，在窗口中单击 Shift + A 即可添加一个输入类结点，在结点的下拉菜单中选择已新建的材质对象，并调节相应的着色参数即可。

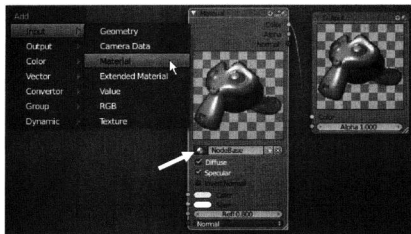


图 10-12 输入类结点菜单和 Material（标准材质）结点

#### 10.2.1.2 Extended Material（增强型材质）结点

Material 是一种简化的材质输入类结点，单击 Diffuse 或 Specular 可开启输入的通道类型，但是这里 Diffuse 和 Specular 的具体参数值需要在材质面板中设置，只能直接调节它的色彩属性。如果需要在结点对输入材质做更直接的调整，可以使用 Extended Material（增强型材质）结点。如图 10-13 所示，它添加了更多的着色选项输入，以及 Diffuse 和 Spec 等通道的单独输出。

#### 10.2.1.3 RGB（色彩）结点

不仅可以使结点上的拾色器来为材质节点选择色彩输入，还可以使用 RGB（色彩）结点，来定义输入的色彩值。RGB 结点的功能很简单，它以一个独立的拾色器面板来控制输出的色彩信息。调节方式同系统中的通用拾色器一样，使用鼠标 LMB 选择合适的颜色即可。如图 10-14 所示，单击 Shift + A 即可添加一个 RGB（色彩）控制结点，按照接口颜色的匹配原理，将其连接至 Color、Spec 和 Mirror 通道等黄色接口中，分别用于控制材质节点的 Diffuse（漫反射）色彩、Specular（高光）色彩和 Mirror（反射光）色彩。



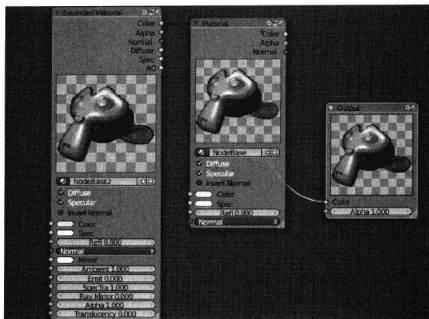


图 10-13 Extended Material (增强型材质) 结点

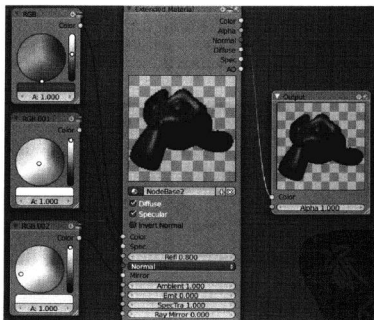


图 10-14 RGB (色彩) 结点

#### 10.2.1.4 Value (数值) 结点

结点上的参数值可以直接进行调节,也可以根据灰色类型的接口,使用数据型的输入结点来控制参数。如图 10-15 所示,为使用 Value (数值) 结点的输入连接效果。Value (数值) 结点没有输入接口,只提供了一个浮点型数值输出,滑动或者直接输入数字即可修改结点的输出值。

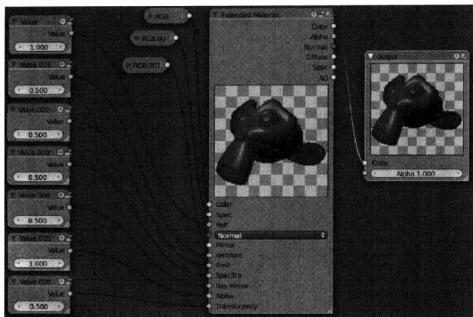


图 10-15 Value（数值）结点示例

#### 10.2.1.5 Texture（纹理）结点

Texture（纹理）结点可将纹理导入结点系统中，它允许附加的 Vector（矢量）坐标输入。如果要贴图映射到物体的 UV 上，则需要在下面的选项框中输入纹理指向的 UV 层，并使用 Geometry 结点输入 UV 的坐标数据。如图 10-16 所示，为一个将纹理映射到材质 UV 坐标上的结点示例图。

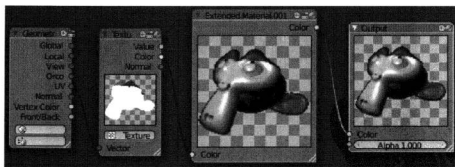


图 10-16 将纹理映射到材质 UV 通道的结点效果

#### 10.2.1.6 Geometry（几何体）结点

Geometry（几何体）结点可输出目标材质或纹理在物体上的空间映射坐标数据，包括 Global（全局）坐标映射数据、Local 镜头与物体表面的矢量映射坐标数据、View（视图）坐标映射数据，其中 Orco（Original Co-ordinates）为默认的纹理映射坐标数据，Normal 为纹理表面的法向坐标映射数据，Vector Color 则用于输出顶点组的色彩。Geometry（几何体）结点常配合 Texture（纹理）结点来制作特殊纹理材质的合成输出。如图 10-17 所示，为使用 Geometry（几何体）结点来定义同一个纹理缩放分布的结点示例。

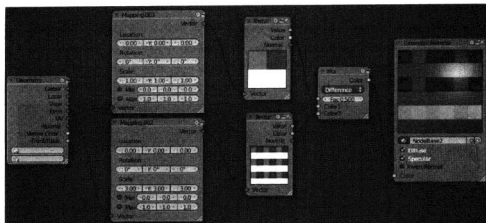


图 10-17 Geometry（几何体）结点的效果

Geometry 结点的 Front/Back 功能也十分强大，它可为物体提供双面的材质蒙板效果。如图 10-18 所示，为一个简单的混合结点设置，最右侧为双面材质的示例。



图 10-18 Front/Back（双面）坐标输出的效果

#### 10.2.1.7 Camera Data（镜头数据）结点

Camera Data（镜头数据）结点是一种输入类结点，它记录了镜头的相关数据，常用于游戏制作中，例如制作随镜头移动时背景的景深效果等。如图 10-19 所示，为一个游戏景深类材质的示例结点。当然也可以使用镜头的 Mist（雾效）功能，来制作这种景深效果。

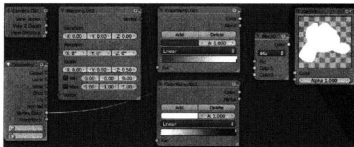


图 10-19 镜头景深材质的示例结点

Camera Data 结点提供了三种镜头数据：Vector（运动矢量），用于显示均匀变化的本地坐标数据；Z Depth（通道景深），用于输出镜头的水平线深度；而 Distance（距离）数据则显示了未被均匀化处理的本地矢量坐标系，也就是镜头与物体的实际距离。首先在场景中添加一个大圣

物体，使用刚才新建的景深材质，并在物体的属性中开启 GLSL 显示模式，以便产生实时的着色计算，如图 10-20 左图所示。接着转入镜头视角，使用 Shift + F 将镜头拉近物体时，物体的材质将显示为结点中 Ramp 参数所控制的颜色，如图 10-20 中图所示。当使用 Shift + F 镜头飞行模式拉开镜头与物体的距离时，可以发现物体材质颜色将随镜头与物体间距离的增大而逐渐过渡为白色，也就是线程中由 Ramp 结点控制的远距离色彩，效果如图 10-20 右图所示。



图 10-20 镜头景深控制物体材质色彩的处理效果

## 10.2.2 输出类结点

材质模式下只有一种输出类结点，就是 Output（材质输出）结点，它用于显示结点系统的合成结果。单击 Shift + A 在 Output 菜单中选择添加即可，如图 10-21 所示。



图 10-21 Output（输出）结点

## 10.2.3 色彩类结点

色彩类结点用于调节材质的色彩属性（包括色彩亮度、对比度和混合模式等），单击 Shift + A 可在 Color 菜单中选择相关的色彩调节结点。如图 10-22 所示，为 Color（色彩）类结点的二级菜单。



图 10-22 Color（色彩）类结点的二级菜单

### 10.2.3.1 Mix（混合）类结点

**Mix（混合）**类结点有3个输入接口，Color1 和 Color2 可分别连接材质或纹理的色彩数据。其中 Color1 为基色，Color2 为混合色彩，结点将根据下拉菜单中的混合模式将 Color2 叠加到 Color1 上。Fac（Factor）系数值为 Color2 的叠加百分比，当值为 0 时，Color2 的输入将不会被混合至输出接口。Mix 结点一共提供了 18 种混合模式，下面将一一介绍。

**Mix：**混合模式，两种输入值将使用 Fac 的数值来产生均匀的混合效果，例如，当值为 0.5 时，两者将各取一半的色彩值做混合计算。使用一个简单的结点示例来演示 Mix（混合）模式，其中接入纹理的 Alpha 通道为混合遮罩，效果如图 10-23 所示。

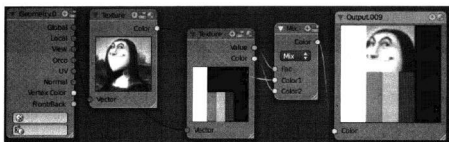


图 10-23 Mix（混合）模式的处理效果

**Add：**添加模式，Color2 将根据 Fac 的数值来决定将多少色彩添加至 Color1 中，例如设置值为 0.5 时，结点将 Color2 添加其 50% 的色彩值在 Color1 上，效果如图 10-24 所示。

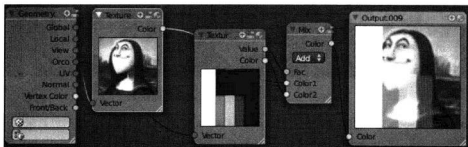


图 10-24 Add（添加）模式的处理效果

**Subtract：**减去模式，用 Color2 减去 Color1 的色彩值，这与 Add（添加）模式的效果相反。如图 10-25 所示，为 Subtract（减去）模式的结点效果。

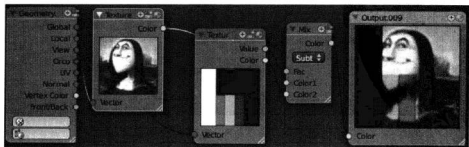


图 10-25 Subtract（减去）模式的处理效果

**Multiply：**正片叠加模式，它将 Color2 的颜色与 Color1 做叠加处理，用于降低 Color1 的亮

度。由于色彩与白色叠加时，混合结果值将不变，而与黑色叠底后将得到黑色，因此中间色在混合后将比任何源色彩都要暗，它与 Screen（屏幕）模式的效果相反。如图 10-26 所示，为 Multiply（正片叠加）模式的结点效果。

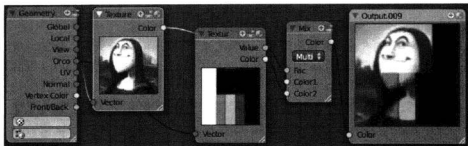


图 10-26 Multiply（正片叠加）模式的处理效果

**Screen:** 滤色模式，它与正片叠加模式的计算结果相反，混合效果是用于提高色彩的亮度。Color1 将根据 Color2 的色彩分布，得到一个更淡更亮的颜色结果，这适合提高色彩中的辉光效果。如图 10-27 所示，为 Screen（滤色）模式的结点效果。

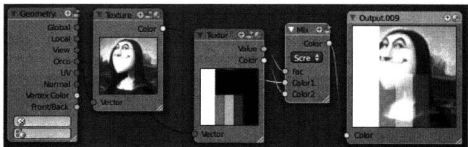


图 10-27 Screen（滤色）模式的处理效果

**Overlay:** 叠加模式，它的效果介于 Multiply 和 Screen 之间，将根据 Color1 的色彩来决定与 Color2 中的哪些色彩以 Multiply 模式来合成，哪些色彩以 Screen 模式合成。合成后的中间色调将产生变亮或变暗的效果，而高光和暗调区域将保持不变。如图 10-28 所示，为 Overlay（叠加）模式的结点效果。

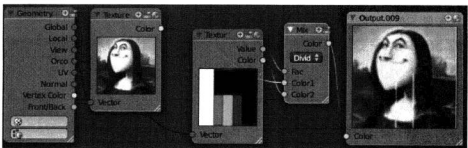


图 10-28 Overlay（叠加）模式的处理效果

**Divide:** 排除模式，与 Difference（差值）模式类似，只是它产生的对比度会较低。这种模式一般比 Difference 更柔和，更明亮，产生更为真实的自然色彩。如图 10-29 所示，为 Divide（排除）模式的结点效果。

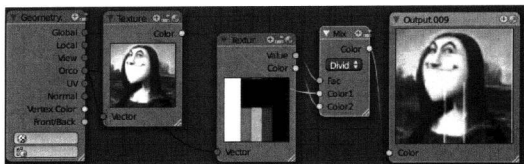


图 10-29 Divide (排除) 模式的处理效果

**Difference:** 差值模式, 它将 Color1 和 Color2 中的色彩做比较, 并用高值减去低值来作为合成后的色彩。所以, 如果使用白色来做差值计算, 将得到反相的效果, 而如果使用黑色则不会发生任何变化。如图 10-30 所示, 为 Difference (差值) 模式的结点效果。

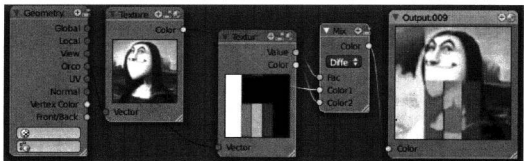


图 10-30 Difference (差值) 模式的处理效果

**Darken:** 变暗模式, 在比较 Color1 和 Color2 的色彩后, 结点将取其中亮度较低的色彩值为混合后的颜色输出, 因此总体的颜色灰度级将偏低, 造成画面变暗的效果。如图 10-31 所示, 为 Darken (变暗) 模式的结点效果。

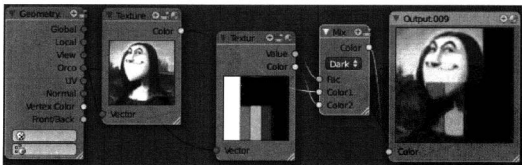


图 10-31 Darken (变暗) 模式的处理效果

**Lighten:** 变亮模式, 这与变暗模式相反, 它取高值为混合后的色彩输出, 总体的颜色灰度等级将偏高, 造成画面变亮的效果。如图 10-32 所示, 为 Lighten (变亮) 模式的结点效果。

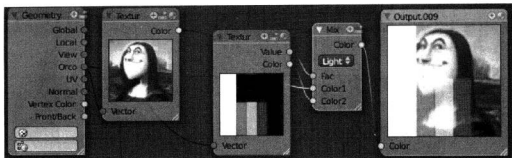


图 10-32 Lighten (变亮) 模式的处理效果

**Dodge:** 减淡模式, 与 Burn 模式相反, 这种模式下结点会通过降低对比度, 加亮 Color1 中的颜色值来反映混合色彩, 其中 Color1 中的颜色越暗, 减淡后的效果越细腻。Dodge 与 Screen 模式类似, 但是前者在 Color2 的边缘区域处理上会显得更尖锐。如图 10-33 所示, 为 Dodge (减淡) 模式的结点效果。

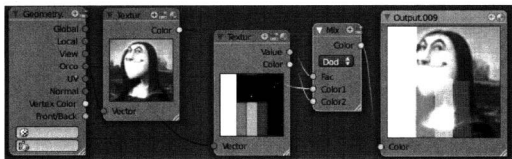


图 10-33 Dodge (减淡) 模式的处理效果

**Burn:** 颜色加深模式, 该模式下结点会忽略 Color1 的色彩值, 叠加中的 Color2 颜色区域越亮, 混合后的效果越细腻。这种方式类似于正片叠底, 但是它会根据叠加的像素颜色, 来适当地增加底层的对比度。如图 10-34 所示, 为 Burn (颜色加深) 模式的结点效果。

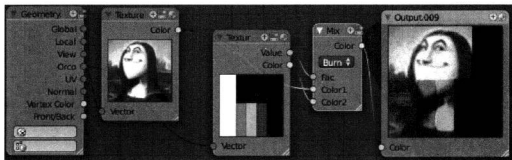


图 10-34 Burn (颜色加深) 模式的处理效果

**Color:** 颜色模式, 它兼顾了 Saturation 和 Hue 两种模式的计算效果, 使用 Color2 的色相值和饱和度, 去替换 Color1 的色相值和饱和度, 但是维持亮度值不变。如图 10-35 所示, 为 Color (颜色) 模式的结点效果。



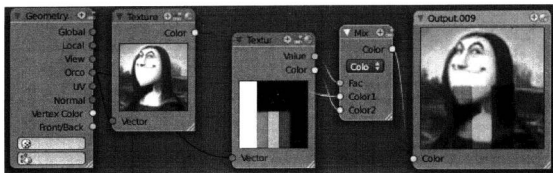


图 10-35 Color（颜色）模式的处理效果

**Value:** 数值模式，Color1 和 Color2 中的 RGB（Red, Green, Blue）输入色彩值将首先被转换为 HSV（Hue, Saturation, Value pixel）数值，在混合它们的饱和度与色相后，再转换回 RGB 数值输出。如图 10-36 所示，为 Value（数值）模式的结点效果。

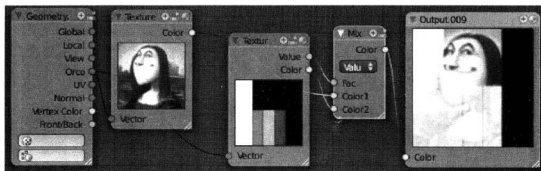


图 10-36 Value（数值）模式的处理效果

**Saturation:** 饱和度模式，结点将使用 Color2 的饱和度去替换 Color1 的饱和度，而色相值与亮度将保持不变。如图 10-37 所示，为 Saturation（饱和度）模式的结点效果。

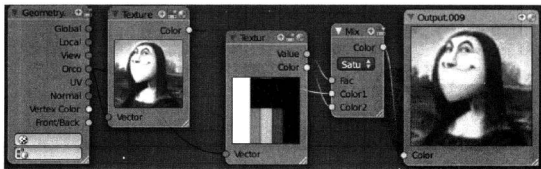


图 10-37 Saturation（饱和度）模式的处理效果

**Hue:** 色相模式，结点将使用 Color2 的色相值去替换 Color1 的色相值，而饱和度与亮度将保持不变。如图 10-38 所示，为 Hue（色相）模式的结点效果。

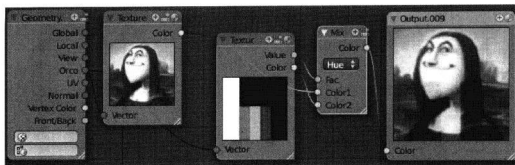


图 10-38 Hue（色相）模式的处理效果

**Soft Light:** 柔光模式，Color2 的输入色彩将以柔光的方式照射在 Color1 层上。合成的结果阶调将趋于画面的中间灰阶调，获得色彩较为柔和的合成效果。如图 10-39 所示，为 Soft Light（柔光）模式的结点效果。

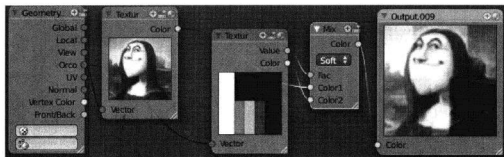


图 10-39 Soft Light（柔光）模式的处理效果

**Linear Light:** 线性光模式，当 Color2 的亮度高于中性灰，也就是 50% 的灰度值时，结点将使用增加亮度的方式调亮 Color1 的色彩值，反之则以降低亮度的方式来调暗输出画面。如图 10-40 所示，为 Linear Light（线性光）模式的结点效果。

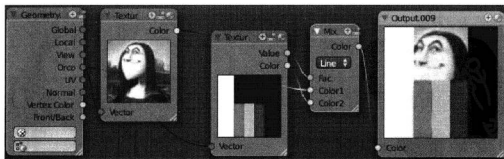


图 10-40 Linear Light（线性光）模式的处理效果

Mix 结点是一种十分重要的合成型结点，也是最常用的结点之一，它可以将多个结点组合在一起，并根据不同的混合模式产生特殊的效果。如图 10-41 所示，为一个简单的结点混合模式示例，分别将两个不同的纹理输入至混合结点的色彩输入通道，并将一个具有黑白色彩值的纹理接入 Fac 通道，通过 Alpha 数据来控制两者的叠加蒙版效果。

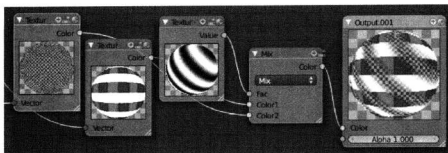


图 10-41 使用纹理来辅助材质合成的处理效果

### 10.2.3.2 RGB Curve (色彩曲线) 结点

Mix 结点仅能调节色彩的混合和叠加模式,为了控制和矫正色彩值,可以使用 RGB Curve (色彩曲线) 结点,来调整输入的色彩值。如图 10-42 所示,为 RGB Curve (色彩曲线) 结点结构。

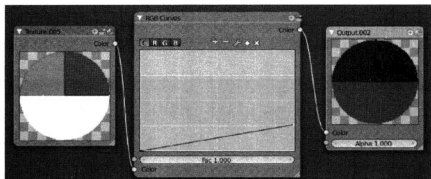


图 10-42 RGB Curve (色彩曲线) 结点的处理效果

色彩曲线结点使用了贝塞尔曲线来调节色彩,其中横坐标为色彩的输入值,纵坐标为色彩的输出值。左上角的 CRGB 选项标识了不同的色彩通道,其中激活 C 将控制全部 Color (色彩) 通道,也可以单击 R、G、B、来分别控制红、绿、蓝 3 个不同色彩的通道值。曲线在空间中的曲率定义了输入色彩与输出色彩的比例值,如果一条曲线的曲率为 1,那么输入值和输出值将不会产生任何变化,色彩以线性的方式输出。单击 LMB 可在曲线上添加控制节点,用来调节这一点的曲率,定义处于这个色彩阶度上的明暗输出比率。

以一个材质调节为例来演示曲线调节的效果。将 (0, 0) 处的点移动至 (0, 1) 位置,那么贴图中最暗的像素点亮度值将被提升 25% 亮度后输出,效果如图 10-43 所示。

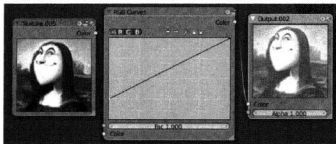


图 10-43 修改曲线的输出系数后的效果 1

同理，也可以改变输入系数，将点从 (0, 0) 移动至 (1, 0)，这时原纹理中 25% 的暗调像素将不再输入至结点中做色彩处理，这样就会造成颜色的细节丢失，效果如图 10-44 所示。

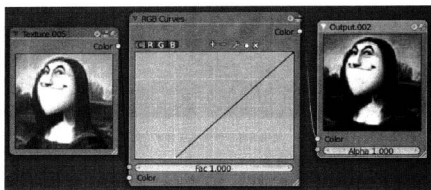


图 10-44 修改曲线的输入系数后的效果 2

当调节线条的中间调时，可以使画面的整体色彩产生加亮或减暗的效果，同时使色彩的饱和度相应地增加或减少。例如采用下弧度曲线时，原纹理的色彩将被均匀加深，产生高对比度的效果，如图 10-45 所示。

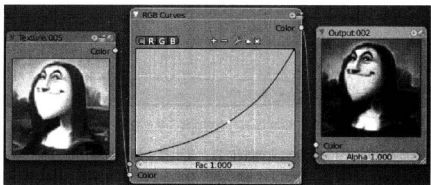


图 10-45 调节曲线的中间曲率的效果

以上调节针对的是 Color 全部色彩通道，也可以选择部分通道来做一些特殊调节。例如当提升红色通道的色彩亮度时，图像整体画面将会显得偏红，效果如图 10-46 所示。

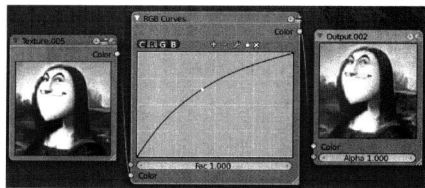


图 10-46 调节单独通道的曲线

### 10.2.3.3 Invert (反转) 结点

Invert (反转) 结点能用于处理一种负片效果, 当值为 0 时, 将关闭结点的处理效果, 当值为 1 时画面将被转换为纯负片, 效果如图 10-47 所示。

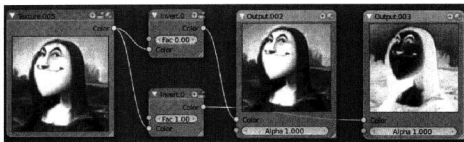


图 10-47 Invert (反转) 结点的效果

### 10.2.3.4 Hue Saturation Value (色相饱和度) 结点

如图 10-48 所示, 为最后一种色彩类 Hue Saturation Value (色相饱和度) 结点。

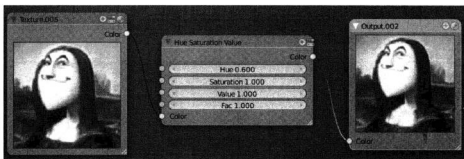


图 10-48 Hue Saturation Value (色相饱和度) 结点的处理效果

**Hue:** 色相调节, 主要用于改变输入的彩色外相, 默认值为 0.5。当减小 Hue 的值时, 画面色彩会发生转变, 蓝色会变得更蓝, 直到转换为绿色和黄色。红色则会渐变为紫色, 最后转换成蓝色和褐色。相反, 如果调高 Hue 值, 将提升画面的红色和绿色, 其中蓝色像素将被转换为紫色, 甚至黄色, 而红色将被转换为金色和绿色。

**Saturation:** 饱和度调节, 用于控制图像色彩的浓淡程度, 默认值为 1。值越低画面的灰度值越高, 当值为 0 时画面将被完全去掉色相值而转换成灰度像素, 如图 10-49 左图所示。当把 Saturation 值调大至 2 时, 色彩浓度将加深, 像素饱和度值被执行二次重叠, 如图 10-49 右图所示。



图 10-49 Saturation (饱和度调节) 的效果

**Value:** 用于调节画面的明度值, 明度值越低, 画面越暗, 当调至 0 时, 画面将变得全黑。而调大 Value 值将增亮画面亮度, 效果如图 10-50 所示。

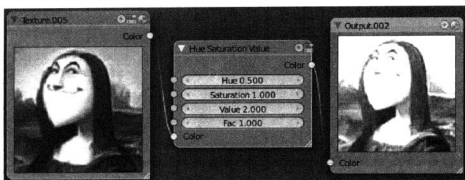


图 10-50 Value (亮度值) 的处理效果

Hue 结点是控制和调节色彩的最佳结点之一, 熟悉一些色彩的变换原则可以有助于更熟练地掌握它的调节。例如蓝色像素无论是将 Hue 设置为最高还是最低值, 它都将被转换为黄色, 如图 10-51 左图所示。同理, 如果是黄色输入值, 无论将 Hue 值设置为最高还是最低值, 它都将被转换为蓝色, 如图 10-51 右图所示。

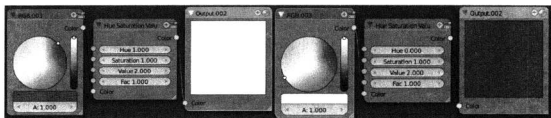


图 10-51 色相环变换示例

## 10.2.4 矢量类结点

### 10.2.4.1 Normal (法向) 结点

矢量类结点用于定义物体表面受光照的影响情况, 如图 10-52 所示, 为使用第一种矢量类 Normal (法向) 结点控制物体表面色彩的结点示例。

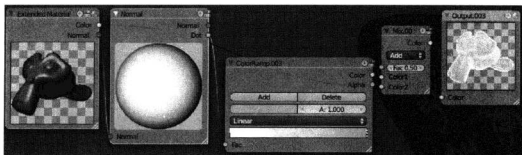


图 10-52 Normal (法向) 结点的处理效果

Normal (法向) 结点可输出一个固定值的 Normal (法向) 数据, 与其他混合类结点配合, 控制物体的法线通道属性。同时, 法向结点也可用于重建材质表面的光照效果, 调节物体在镜头

前的光照角度，只需要使用 LMB 移动灯光照射在材质球上的角度即可。

Normal（法向）通道有两个输出接口，其中 Normal 接口用于输出固定光照方向下的物体法向数据。如图 10-53 所示，使用一个固定的光照角度来自定义地控制纹理的映射模式。

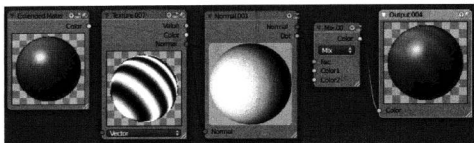


图 10-53 使用 Normal 接口输出光照角度的效果

Dot 接口则用于计算输入的法线数据被光照叠加后的效果，可以用来处理 Geometry 的 Orco（全局光照角度）数据。如图 10-54 所示，首先将 Orco 的坐标数据输入至 Normal 结点，改变其受光照的角度，然后使用 Dot 接口输出并对光照位置执行着色计算，在输出结点上可以看到，材质呈现出了特殊的光照色彩效果。

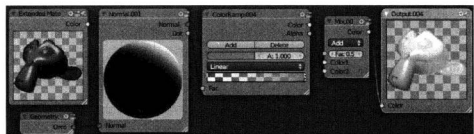


图 10-54 使用 Dot 来计算光照角度的效果

#### 10.2.4.2 Mapping（映射）结点

Mapping（映射）结点，可用于调整输入纹理或材质的映射方式。如图 10-55 所示，为使用 Mapping（映射）结点调整纹理结构的结点示例。

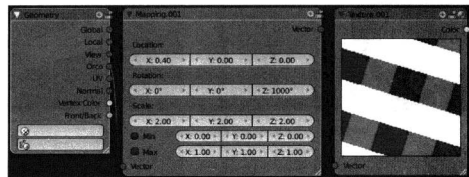


图 10-55 Mapping（映射）结点的处理效果

Location（位移）用于调整纹理映射在各坐标轴上的位移偏差，Rotation 则用于调整纹理的旋转角度，Scale 控制了尺寸的缩放。当 Min 和 Max 选项被激活时，结点将约束纹理坐标输出的最

小值和最大值，其数值可在旁边的参数框中直接输入。

也可以配合 Vector Curves（矢量曲线）节点来灵活地调整坐标的对应映射参数，实现非线性的映射效果。如图 10-56 所示，为配合 Vector Curves（矢量曲线）节点的调节效果。其中左上角的 X、Y 和 Z 选项分别用于切换不同坐标轴上的映射偏移量，曲线的调节原理与方式和 Color Curve 的调节方式相同。

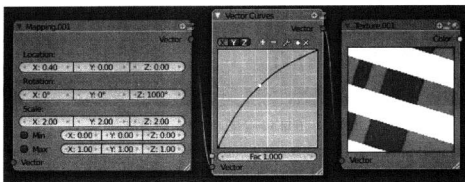


图 10-56 Vector Curves（矢量曲线）节点的处理效果

## 10.2.5 转换类结点

结点的通道不仅支持传统的 RGB 格式和 HSV 格式，还支持 HDMI（High Definition Media Interface）等最新色彩格式的转换。因此常使用 Converter（转换）类结点，来完成色彩的格式转换和通道分离等数据类型转换计算。

### 10.2.5.1 ColorRamp（渐进色）结点

ColorRamp（渐进色）结点是一种色彩处理结点，它只有一个输入端，可输入 RGB 等多色彩数据，也可处理单通道色彩。如图 10-57 所示，为使用 ColorRamp（渐进色）结点做自定义着色的结点示例。

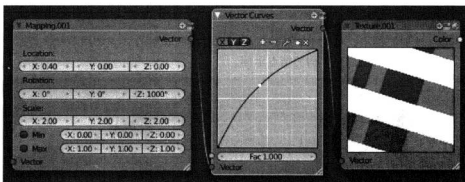


图 10-57 Color Ramp（渐进色）结点的处理效果

渐进色结点的着色原理，是使用色带上的新增色彩来替代输入材质上同比例明度位置的色彩，色带的默认分布是从左至右方向，为最深色至最浅色的渐变。所以，当将最左边位置的颜色设置为黄色时，输入纹理中颜色最深的部分都将被替换为黄色。

色带的修改方式很简单，单击 Add 即可在色带上添加一个色彩控制点，使用 LMB 选择控制点并调整它们在着色区域上的位置。色带下面的色彩调板用于选择当前结点的颜色，其中



A 为该点位置的色彩 Alpha（透明）值，用于控制色彩在色带着色时产生的遮罩输出效果。要删除某个控制点，只需要在选择的状态下按 Delete（删除）按钮即可。

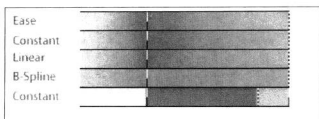


图 10-58 色带显示模式

色带提供了多种渐变模式，用于控制色彩间的过渡效果，包括 Ease（平缓）、Cardinal（鲜艳）、Linear（线性化）、B-Spline（曲线化）和 Constant（常值）模式。如图 10-59 所示，为 Constant（常值）模式下的着色结点效果示例。

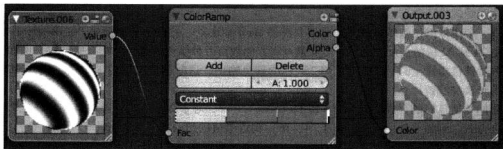


图 10-59 修改结点着色渐变模式的处理效果

#### 10.2.5.2 RGB to BW（去色）结点

RGB to BW（去色）结点功能十分简单，就是将 RGB 的色彩模式转换为黑白模式。如图 10-60 所示，为 RGB to BW（去色）结点的处理效果示例。

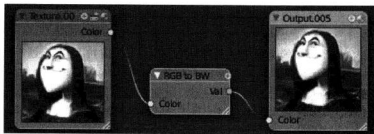


图 10-60 RGB to BW（去色）结点的处理效果

#### 10.2.5.3 Math（数学计算）结点

Math（数学计算）结点是一种公式型数据处理结点，它提供了几乎大部分数学计算方法。Math（数学计算）结点有两个输入端，它们将按照面板中选择的数学计算公式来完成数据的合成与输出。如图 10-61 所示，为使用 Math（数学计算）结点将 Red（红色）通道加上了 0.5 再输出的单通道色彩结点示例。其中配合使用了矢量类结点的 Separate RGB（色彩分离）结点和 Combine RGE（色彩合并）结点，分别用于分离和合并色彩的各通道数据。

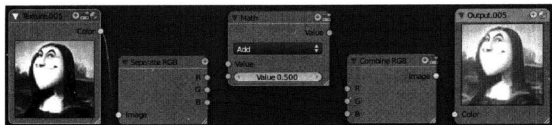


图 10-61 使用 Math（数学）结点处理图像的色彩通道效果

Math（数学）结点的数学计算公式包括 Greater Than（大于）、Less Than（小于）、Round（四舍五入）、Maximum（最大值）、Minimum（最小值）、Logarithm（对数）、Power（幂计算）、Arctangent（反正切）、Arccosine（反余弦）、Arcsine（反正弦）、Tangent（正切）、Cosine（余弦计算）、Sin（正弦计算）、Divide（除法）、Multiply（乘法）、Subtract（减法）和 Add（加法）等运算公式。所有的图像数据都可采用这些公式来进行计算和处理，在材质模式下，结点的主要输入为图像的色彩数据。

以一个图像的位数转换计算为例来讲解 Math（数学）用法。一般来说，在 Blender 中的色彩都是 24 到 32 位的深度图像，每个像素点将存储数百万种颜色信息。但在有些时候，例如游戏制作等特殊色彩输出时，仅需要使用 256 色的颜色输出，这时就可以利用 Math 数学结点来完成色彩的转换和滤镜计算。

我们知道色彩的合成公式为  $Q = \text{Red} \times \text{Green} \times \text{Blue}$ ，对于 256 色的色彩输出有多种色彩比例的组合方式，例如  $[R, G, B] = [4, 4, 16]$  和近似值的组合  $[R, G, B] = [6, 6, 7]$ ，为了保证色彩的输入与输出相互平衡，一般选择后者为滤镜标准。

对于 24 位图像，由于其色彩的数量十分丰富，使得色彩的过渡呈现连续性。而 256 色图像由于色彩数量的局限性，在明暗过渡时会因为色彩数量有限而出现高落差的色彩渐变。例如，如图 10-62 所示，从红色过渡到黑色时，24 位的色带可能会存在几百种色彩，其渐变效果十分平滑，而 256 色模式下只拥有 6 种颜色，其色彩过渡十分生硬。为了处理这种对色彩的分类和过滤，在计算过程中需要将一系列连续的彩带根据区间划分替换为 6 种独立的数字色彩，例如将值为 0 和 0.16 之间的颜色都统一替换为值是 0 的颜色。

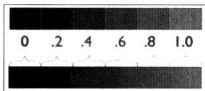


图 10-62 256 色下对过渡色彩的划分

那么，如何将这些连续数字执行范围界定计算呢？由于最终值只允许均匀的 6 组数据，所以可以首先将数值按比例放大，减去区间值的一半，再使用四舍五入函数，使数值的范围缩小至 0 至 5 之间，如果此时要再转换回 0 至 1 之间，只需要除以区间的数量即可。使用公式来描述这个算法就是  $f(x, n) = \text{Round}\left(x \times n - \frac{1}{2}\right) / (n-1)$ ，其中， $x$  为输入数据， $n$  为色彩的区间数量。但是，这个公式在处理部分特殊数据时会出现数据异常，例如当输入值为 1 时，计算结果将为 1.2，超出我们的界定范围。而当值为 0 时，Round 函数也将因为无法返回负值而出现运算结果丢失。但是由于这样的极端数据量比较少，所以可以简单地忽略不计。

在完成了理论学习和算法设计后，就可以使用 Math（数学）结点来实现这种色彩的转换过程了。对 RGB 转换时，可以将三种色彩通道分离出来，然后对每种通道按照  $[R, G, B] = [6, 6, 7]$  的比例做筛选处理。例如：一个 0.7 的 R 通道数据，在乘以 6 后将得到 4.2，减去 0.5 后的结果为 3.7，接下来再使用 Round 做四舍五入操作后可以得到一个近似

值4,最后再将4除以5即可得到替代色彩的数值为0.8,色彩的转换完成。如图10-63所示,为实现这一算法的结点地图示例。当把整个中间结点都打包成组后,即保存为一组特殊的色彩处理滤镜结点。



图 10-63 256 色位图转换结点的处理效果

#### 10.2.5.4 Vector Math (矢量数学) 结点

不同于 Math (数学) 结点, Vector Math (矢量数学) 结点主要用于计算矢量数据。矢量数据是一种有大小、有方向的数学表达式,在 Blender 中常用的矢量数据为 Normal (法向) 值。如图 10-64 所示,为根据 Vector Math (矢量数学) 结点,来计算材质法线并控制其混合模式的结点示例。

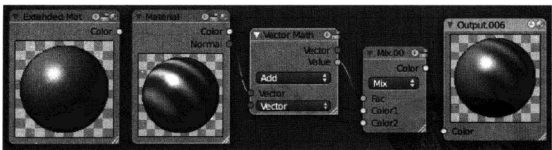


图 10-64 Vector Math (矢量数学) 结点的处理效果

Vector Math 包含了两个输出接口,其中 Vector 仅用于输出矢量型数据。而 Value 不仅能输出数值型数据,还可包含额外的数学计算效果。Vector Math 提供了多种计算函数,包括 Add (加法) 计算、Subtract (减法) 计算、Average (平均值) 计算、Dot Product (点乘积) 计算、Cross Product (矢量积) 计算和 Normalize (标准化) 计算。其中 Dot Product (点乘积) 计算是一种标积计算,它返回的是数值型结果,而 Cross Product (矢量积) 计算是一种向量积计算,它返回的值是矢量型数据。如图 10-65 所示,为使用 Vector Math 的 Normalize (标准化) 计算,将法线的矢量分布信息描绘出来的结点示例。

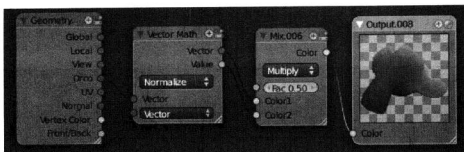


图 10-65 Vector Math 的 Normalize (标准化) 计算效果

#### 10.2.5.5 Squeeze Value (数值挤压) 结点

Squeeze Value (数值挤压) 结点是最后一种材质类计算结点, 它主要与 Camera Data (镜头数据) 结点搭配使用。当 Camera Data 的输出值较大, 远远超过使用范围时, 就可以用 Squeeze Value 结点将这些大数值的景深等数值等比例转换至可接受的范围内。如图 10-66 所示, 为 Squeeze Value (数值挤压) 结点的一个比例原理图, 横坐标为数据的输入值, 纵坐标为结点的输出值, Center 值为输入数据的转换基点, 这一点的值在输出时将被定义为 0.5, 而其他数据则将以此为原点向两极按等比例扩展, 但是 Squeeze Value 的输出值永远在 0 到 1 区间之内。Width 参数定义了转换曲线的曲率, 值越小则曲线越趋于线性。可以看到红色的曲线是 Width 值为 0.2 时的输出曲线, 它相比蓝色 Width 值为 1 时的曲线显得更平缓。

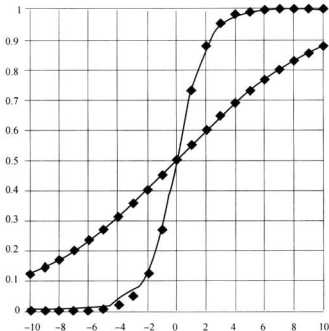


图 10-66 Squeeze Value (数值挤压) 结点的原理图

如图 10-67 所示, 为一个基本的 Squeeze Value (数值挤压) 结点的应用示例。当物体距离镜头为 10 个 Blender 单位时, 材质将显示为黄色, 而当距离越远时物体的材质颜色将呈现为白色; 相反, 离镜头越近物体则将显示为黑色。

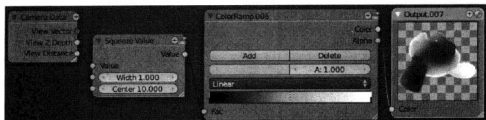


图 10-67 Squeeze Value（数值挤压）结点的处理效果

## 10.3 纹理类结点

Texture（纹理）类结点能使用结点的多线程处理特性，快速地制作出一些多组合的特殊 Texture（纹理）效果。使用前需要首先在结点编辑器中将处理模式切换至纹理模式，同时激活 Use Nodes 按钮，如图 10-68 所示。



图 10-68 激活纹理的结点模式

同结点类材质的使用方法一样，可以在状态栏中命名纹理结点的名称，单击加号可新建一个结点类纹理。它与结点类不同的地方在于，纹理类结点不会显示在 3D 视图窗口中或者游戏引擎中，它只会显示在最后的渲染结果中。

### 10.3.1 输入类结点

#### 10.3.1.1 Time（时间）结点

第一种输入类结点为 Time（时间）结点，这是一种用于制作动画纹理的结点，控制着输出纹理随时间的变化而产生的动画效果。如图 10-69 所示，为一个由时间帧数控制纹理色彩的示例。

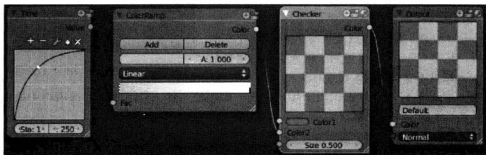


图 10-69 Time（时间）结点控制纹理色彩的处理效果

Time（时间）结点没有输入接口，它只有一个由曲线控制的区域和帧数设置选项。它的输出值为 0 至 1，数值由当前曲线时间点上的纵坐标决定，而横坐标则为时间轴坐标，曲线的控制和 Blender 系统中的曲线控制器类似。如图 10-70 所示的结点示例，当渲染第 0 帧时，输出值为 0，这使得图案色彩以 ColorRamp 中最小值时的颜色，也就是黄色来输出，如图 10-70 左图所示。当将帧移动至第 250 帧时，输出控制值为 1，此时的图案色彩将以 ColorRamp 中最大值定义的白色来输出，效果如图 10-70 右图所示。

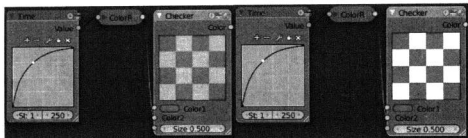


图 10-70 使用 Time (时间) 结点的输出值来控制色彩的效果

### 10.3.1.2 Coordinate (坐标) 结点

Coordinate (坐标) 结点提供了纹理的坐标系数数据, 它没有任何输入控制参数, 所有的坐标选项都在纹理面板中的 Map Input (映射输入) 面板中。如图 10-71 所示示例结点, 可以使用 Coordinate (坐标) 结点来辅助实现纹理坐标上的着色操作。



图 10-71 Coordinate (坐标) 结点输出物体的坐标数据

在示例结点中, 输出纹理坐标的 XYZ 数据, 并使用 Decompose RGB 通道分离结点来对每个坐标轴分别执行着色操作。最后对按照 Z 坐标深度做着色处理的纹理执行渲染操作, 效果如图 10-72 所示。

### 10.3.1.3 Texture Image (图片纹理) 输入结点

可以向结点系统自定义输入纹理和图片素材, 如图 10-73 所示, 为 Texture (纹理) 输入结点和 Image (图片) 输入结点。

Texture (纹理) 输入结点允许导入任意已存在的纹理素材, 其中 Color 端接口可用于输入纹理的覆盖色彩, Texture 结点不支持 Alpha 数值的输入。

Image (图片) 输入结点则仅用于导入系统中的图片文件, 只需要单击选项栏中的浏览按钮, 即可选择系统中的图片素材。



图 10-72 对分离出的坐标执行着色的效果

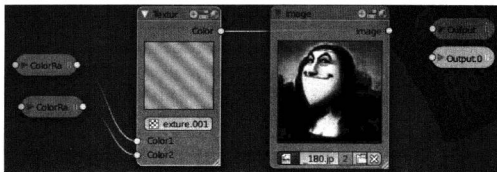


图 10-73 Texture (纹理) 输入结点和 Image (图片) 输入结点

### 10.3.2 输出类结点

输出类结点用于显示纹理的计算结果,如图 10-74 所示,为 Output (输出) 结点和 Viewer (预览) 结点。



图 10-74 Output (输出) 结点和 Viewer (预览) 结点

Output (输出) 结点显示了最终的运算结果输出,在下面的输入框中键入命名,即可在纹理面板中直接调用当前的结点纹理了。Normal (法向) 输入接口用于定义表面的法向参数,使纹理能被正确地应用于 Bump (凹凸) 贴图或 Normal (法向) 贴图中。

Viewer (预览) 结点仅用于预览合成的任意中间过程,只能显示贴图的色彩信息。

### 10.3.3 色彩类结点

纹理类的色彩结点和材质类的色彩结点类似,只是这里新增了两种通道分离结点——Compose RGBA (色彩分离) 结点和 Decompose RGBA (色彩合成) 结点,如图 10-75 所示。

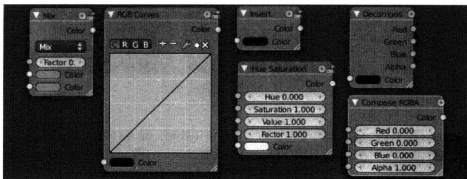


图 10-75 Color 色彩类结点

Compose RGBA (色彩分离) 结点和 Decompose RGBA (色彩合成) 结点的功能十分简单,前者可将纹理的各通道分别分离,而后者则是将各通道合并在一起。

### 10.3.4 图案类结点

Pattern (图案) 类结点能快速地生成纹理图案,这类结点默认拥有两种图案,Checker (检查器) 图案结点和 Bricks (砖块) 图案结点。

#### 10.3.4.1 Checker (检查器) 图案结点

Checker (检查器) 图案结点是一种默认的网格图案,它只有两种颜色,常用于纹理的拉伸比例检测,结点的 Size 选项用于控制网格的尺寸大小。如图 10-76 所示,为 Checker (检查器) 图案结点。

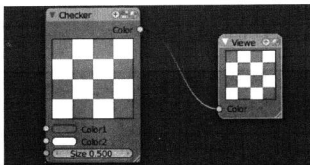


图 10-76 Checker（检查器）图案结点

#### 10.3.4.2 Bricks（砖块）图案结点

Bricks（砖块）图案结点可快速地生成墙体的砖块效果，如图 10-77 所示。其中 Offset 参数定义了砖块间的偏移量，而 Frequency 则用于控制砖块间的变换间隔数量。Bricks 颜色可分别选择两种砖块的颜色，而 Mortar 则用于定义砖块间缝隙的颜色。

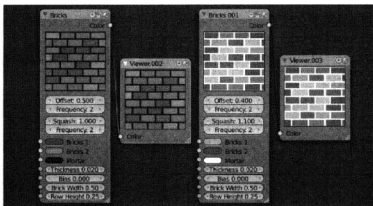


图 10-77 Bricks（砖块）结点

#### 10.3.5 内置纹理结点

Texture（内置纹理）结点同输入结点中的 Texture（纹理）结点区别在于，后者可导入系统中已执行过编辑的纹理对象，而前者则用于导入纯内置纹理。如图 10-78 所示，为纹理结点菜单下全部类型的结点效果。

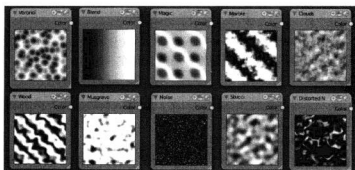


图 10-78 Texture（纹理）类结点



### 10.3.6 转换类结点

纹理的转换类结点除了包括与材质类结点功能相同的 Math 数学计算结点、ColorRamp（色彩渐变）结点、RGB to BW（去色）结点，还有两个新增结点，它们分别是 Value to Normal（法线数值转换）结点和 Distance（距离）结点。

#### 10.3.6.1 Value to Normal（法线数值转换）结点

Value to Normal（法线数值转换）结点用于将数值数据或色彩数据转换为纹理的法线贴图数据。如图 10-79 所示，为 Value to Normal（法线数值转换）结点的应用示例。

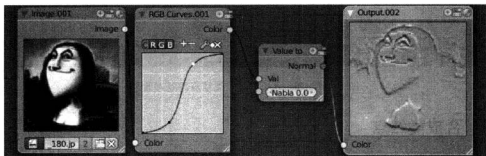


图 10-79 使用 Value to Normal（法线数值转换）结点来制作法线贴图的效果

Value to Normal（法线数值转换）结点有两个输入接口，其中 Value 用于输入色彩信息，而 Nabla 则用于控制法线纹理的线条深度，其值越大纹理线条的深度值越大，取值范围为 0 至 0.1。

#### 10.3.6.2 Distance（距离）结点

Distance（距离）结点用于计算固定点到物体纹理坐标之间的距离。如图 10-80 所示，为将坐标距离做着色处理的结点示例。

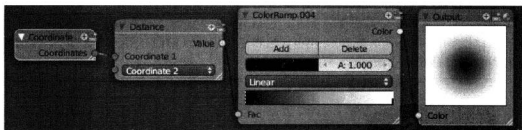


图 10-80 Distance（距离）结点的处理效果

Distance（距离）结点的输入为坐标轴的数据，可以使用 Coordinate（坐标轴）结点导入当前物体的坐标数据，也可单击下拉菜单，手动输入运算点的坐标数据。

### 10.3.7 扭曲类结点

Distort（扭曲）类结点用于修改纹理的形变属性，包括 Rotate（旋转）结点、Translate（形变）结点、Scale（缩放）结点和 At（坐标轴重定义）结点。如图 10-81 所示，为各扭曲类结点对纹理的修改情况。

扭曲类结点的输入接口均为纹理的色彩数据，分别使用各自的参数设置来调节纹理形变状态。其中 Rotate（旋转）结点可使用 Turn（转动）参数来控制纹理的旋转角度，Translate（形变）结点则使用 Offset（偏移）参数来更改纹理的位移变化，而 Scale（缩放）结点中的 Scale

(缩放) 参数则可调整纹理的大小比例。At 坐标轴结点较特殊, 它能重定义纹理的坐标原点中心, 使用 Coordinate (坐标数据) 来输入即可。

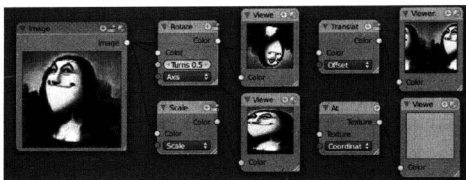


图 10-81 Distort (扭曲) 类结点的处理效果

## 10.4 合成类结点

合成类结点用于对渲染结果执行后期处理, 渲染结果包含了多种数据信息, 后期结点能对其中不同的数据做放大或滤镜等效果处理, 如同类似的后期工具 Photoshop 和 Gimp 一样。处理前需激活后期结点模式, 在结点编辑器状态栏中切换至合成图标, 并激活 Use Nodes (使用结点) 选项, 如图 10-82 所示。




图 10-82 激活后期合成结点功能

### 10.4.1 输入类结点

合成类结点允许更多的数据输入方式, 包括 Render Layers (渲染图层)、Image (图片)、Texture (纹理贴图)、Value (数值)、RGB (色彩) 和 Time (时间) 等。如图 10-83 所示, 为 Render Layers (渲染图层) 结点。



图 10-83 Render Layers (渲染图层) 结点

渲染图层结点没有输入接口，它将根据渲染中设置的渲染图层属性，直接将渲染结果导入至结点系统。由于每个 Blender 工程可包含多个 Scene（场景），因此需要首先选择场景，再选择场景下设置的输出图层。渲染结点的数据输出选项十分丰富，可根据渲染面板中对渲染层的设置开启对应图层的输出通道。当可能需要对场景中的某一个图层单独重新渲染更新时，不需要单击 F12 来渲染全部图层，只需要单击结点上的  图标即可仅对当前图层执行渲染更新操作。

合成模式下的 Image（图片）结点不仅能输入单纯的图片文件，还可输入视频文件，如图 10-84 所示。



图 10-84 Image（图片）结点

Image（图片）结点的 Source（来源）选项提供了 4 种文件输入格式。File（文件）可选择外部的图片文件，Sequence（序列）文件和 Movie 选项可用于导入视频文件，前者导入的是一连串连续的文件序列，而后者则用于导入单独的完整视频文件。这里的播放帧率设置应与渲染面板中的回放帧率相同，否则系统将会默认使用回放帧率来播放素材视频。Generated 选项用于输入在 UV Editor（编辑器）下制作或生成的图片文件。

对于 Texture（纹理贴图）、Value（数值）、RGB（色彩）和 Time（时间）结点，请参考材质和纹理模式下的它们的使用方式。

## 10.4.2 输出类结点

### 10.4.2.1 Composite（合成效果）结点

Output（输出）类结点用于导出最后的合成效果，以及预览合成的中间效果。如图 10-85 所示，为 Composite（合成效果）结点。

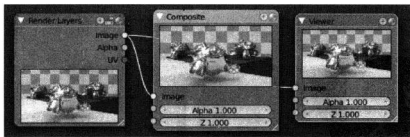


图 10-85 Composite（合成效果）结点

Composite（合成效果）结点用于输出最后的合成结果，其中 Image 接口用于输入图像数据，而 Alpha 和 Z 接口则分别用于输入 Alpha 通道数据和 Z 通道数据。结点系统中仅允许使用一个 Composite（合成效果）结点，在完成渲染后使用快捷键 F3，即可将结点的输出结果保存为本地文件。

### 10.4.2.2 View（预览）结点

预览结点有两种模式，分别为 View（预览）结点和 Split Viewer（拆分预览）结点。前者可用于预览完整的输出画面，而后者则可将两个输出画面合并在一起，如图 10-86 所示。



图 10-86 View（预览）结点和 Split Viewer（拆分预览）结点

Split Viewer（拆分预览）结点可选择按照 X 轴切分或者按照 Y 轴切分，其中 Factor（系数值）决定了两个输入画面所占预览图像的比例大小。通常情况下，在比较多场景间的输出结果时，可使用拆分预览结点来比较输出的结果。

#### 10.4.2.3 File Output（文件输出）结点

File Output（文件输出）结点用于存储合成的输出图像，如图 10-87 所示，为 File Output（文件输出）结点。

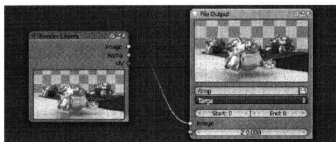


图 10-87 File Output（文件输出）结点

文件输出结点的输入端为需要保存的图像数据，在结点的参数面板中可以选择保存的路径和保存的文件格式。如果渲染输出的是视频序列，可以使用 Start 和 End 来控制保存的序列帧范围。

#### 10.4.2.4 Levels（灰度图）结点

Levels（灰度图）结点用于显示当前图像的灰阶直方图，如图 10-88 所示，为 Levels（灰度图）结点的输出效果。

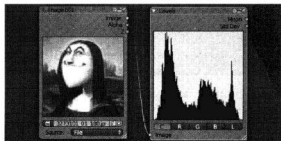


图 10-88 Levels（灰度图）结点的显示效果

### 10.4.3 色彩类结点

#### 10.4.3.1 Bright/Contrast（明度/对比度调节）结点

合成类色彩结点与材质结点可共用 RGB Curves（色彩曲线）结点、Mix（混合类）结点、

Hue Saturation Value (色相饱和度) 结点和 Invert (反转) 结点,并新增了7种调节类结点。如图 10-89 所示,为 Bright/Contrast (明度/对比度调节) 结点。

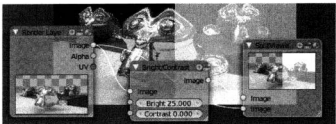


图 10-89 Bright/Contrast (明度/对比度调节) 结点的处理效果

Bright/Contrast (明度/对比度调节) 结点的输入接口为图像数据,参数面板中的 Bright (明度) 可用于调节画面的亮度,值越高画面的亮度越高。而 Contrast (对比度) 值增加,画面中的浅色区域将被高亮,使阴暗区域被加深。

#### 10.4.3.2 Gamma (伽玛) 值结点

前面曾讨论过,在图像制作中,由于显示器本身的物理结构差异,和其他色彩同人眼之间的色彩误差,打印输出的结果色彩可能会与我们在显示器上看到的色彩存在偏差。对输入输出值所做的 Gamma 值修正过程,被称作 LWF (Linear Workflow) 线性工作流。

显示器的输出色彩可以用一个公式来表达,  $Output = Input^{Gamma}$ , 不同显示器的 Gamma 值均不同,例如 Mac 为 1.8,而普通的 CRT 为 2.2。因此屏幕上实际的输出结果就是一种 Non-linear Display (非线性) 的显示输出,这样的输出结果很难保证协调一致的色彩。所以当把在 PC 上制作的图像移植到 Mac 上查看时,就会发现画面会略微显得灰白。

校正画面 Gamma 的方式有很多,硬件层包括更换显示器和操作系统,在软件 Blender 中可以使用 Gamma 结点来调节画面输出。首先在渲染面板的 Shading (着色) 选项中开启 Color Management (色彩管理) 功能,如图 10-90 所示。这样做的目的是启动纹理和材质等色彩的输入 Gamma 预校正,也就是说将输入色彩信息在读取至内存前,就先匹配至显示器的非线性特性,这样在材质编写和合成时,从显示器上看到的色彩就和实际色彩相吻合了。

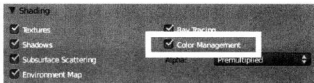


图 10-90 开启 Color Management (色彩管理)

校正的过程是先将输入取 Gamma 为 2.2 的叠加计算,系统将根据输入值做一个查色反馈,并为其指派一个反相 Gamma 曲线使色彩在显示器上可以正确显示。经过这样的转换,显示色彩本身又被进行了一次 Gamma 为 0.4545 的校正计算,使校正后的 Gamma 显示结果为 1。

图像信息在系统内部将执行 Gamma 为 1 的色彩数据计算,也就是说,此时系统内将工作在线性色彩空间下,而过程画面由于二次校正将配合显示在已被校正的显示器屏幕上,视觉上工作在与肉眼匹配的线性色彩空间。当完成渲染和合成工作后,还需要对输出图像再做一次二次校正,还原为与输入数据相同的色彩空间,并保存在文件信息中。如图 10-91 所示,为将输出结果连接一个 Gamma 为 0.4545 的结点,做二次输出校正的处理效果。

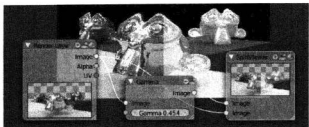


图 10-91 对输出做二次 Gamma 校正

#### 10.4.3.3 Alpha Over (透明重叠) 结点

Alpha Over (透明重叠) 结点可根据图片的 Alpha 透明通道信息来做合成处理, 如图 10-92 所示, 为 Alpha Over (透明重叠) 结点的结点合成示例。

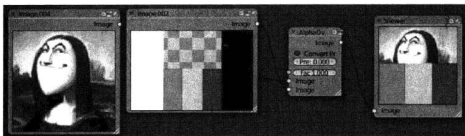


图 10-92 Alpha Over (透明重叠) 结点的处理效果

Image1 的接口输入为合成结果的背景图, 而 Image2 的输入图像需要拥有 Alpha 值大于 1 的透明蒙版通道, 使得结点产生透明叠加效果。Factor 值控制了叠加图层的透明度, 当 Convert Premu 开启时将激活边缘的平滑效果, 降低由于 Alpha 通道在临界位置产生的锯齿效果。

#### 10.4.3.4 Z Combine (深度通道合成) 结点

Z Combine (深度通道合成) 结点能将不同场景中的不同物体, 按照镜头与物体间的深度值来做混合运算。如图 10-93 所示, 为 Z Combine (深度通道合成) 结点合成两个场景中物体的结点示例。参与混合的图像必须存在于不同的场景, 它们之间的合成深度也由各自场景中的镜头景深值来决定。

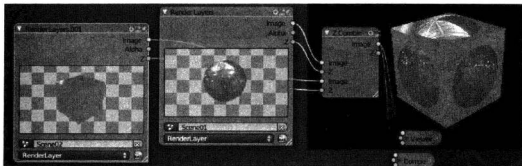


图 10-93 Z Combine (深度通道合成) 结点合成多场景间物体的处理效果

#### 10.4.3.5 Color Balance (色彩平衡) 结点

Color Balance (色彩平衡) 结点使用符合 ACL 规范的色彩来分级修正画面的曝光, 从概念

上讲, 图像的彩色包括暗部、中间层次和亮部三个部分。色彩平衡结点中可使用 Lift (密度) 值来控制色彩的暗部, 用于强调阴影, 使用 Gamma (伽玛) 值来调整中间调, 而利用 Gain (增益) 值来调节亮部的高光部分。如图 10-94 所示, 为 Color Balance (色彩平衡) 结点的调节示例。

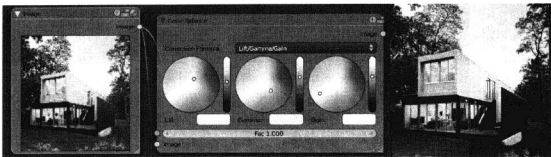


图 10-94 Color Balance (色彩平衡) 结点的处理效果

Color Balance 不仅支持分级修正, 还支持由美国电影摄影师协会制定的色彩确定表 ASC-CDL (The American Society of Cinematographers-Color Decision List) 修正, 它是一种标准的色彩修正平台, 其目的是将色彩校正信息存储在发行介质上。这种模式下将分别使用 Offset、Power 和 Slope 参数来调节图像的暗部、中间层次和亮部。

#### 10.4.3.6 Hue Correct (色相校正) 结点

Hue Correct (色相校正) 结点是一种不同于 Hue Saturation Value (色相饱和度) 的结点, 它采用曲线的方式, 按照 HSV 色彩阶度来调节色相中的色彩深度。如图 10-95 所示, 为 Hue Correct (色相校正) 结点图像校正的示例。

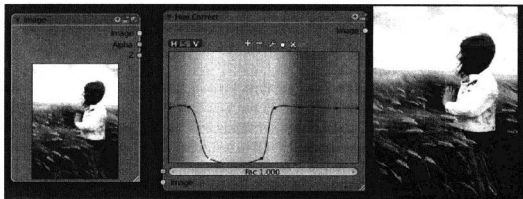


图 10-95 Hue Correct (色相校正) 结点的处理效果

色相校正中的曲线分布代表了色彩在画面中的比重, 曲线点在 Y 轴的位置越高表示色彩的深度越高, 当移动点至底部时, 这个位置所代表的色彩将从画面中被去除掉。

#### 10.4.3.7 Tonemap (色调调和) 结点

Tonemap (色调调和) 结点用于处理 OpenEXR 等 HDR 高动态文件格式。由于 OpenEXR 文件的色彩高动态和无压缩等特性, 传统的显示设备无法正常查看超出 24 位的图像亮度等级, 可以使用 Tonemap (色调调和) 结点来将 HDR 图片转换为正常的可视文件。如图 10-96 所示, 为一个处理曝光不足的结点示例。

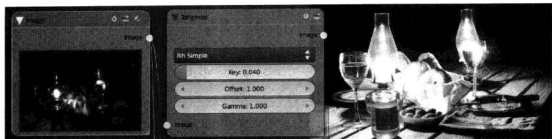


图 10-96 使用 Tonemap（色调调和）结点来转换 HDR 高动态文件

Tonemap（色调调和）结点并不是一种用于增加艺术表现力的滤镜结点，而是针对 HDR 文件存储的高密度光子数据，使用两种转换算法 R/D photoreceptor 和 Rh Simple，分别将动态范围内的光子数据转换为线性可视化的显示输出。

### 10.4.4 矢量类结点

#### 10.4.4.1 Normal（法向）结点

在合成类结点中，Normal（法向）结点的用法和材质合成中略有区别。可以使用 Normal（法向）结点按照物体的 Normal（法向）通道，为渲染结果添加额外的全局光照效果。如图 10-97 所示，为 Normal（法向）结点的应用示例。



图 10-97 Normal（法向）结点的处理效果

#### 10.4.4.2 Map Value（映射值）结点

使用 LMB 来移动 Normal 上的球光影，改变物体表面的光照角度，用于调节全局光对物体法向上光照的影响。同时也可以使用矢量结点中的 Map Value（映射值）结点，来调节法线的深度值。这里的 Offset（偏移量）可为输入值叠加一个固定值，而 Size（增益）值则用于控制输入值的映射强度，Use Minimum/Maximum 选项用于设置结点输入极值范围。如图 10-98 所示示例，可以使用多个 Map Value（映射值）结点来完成一些特殊数值输出。

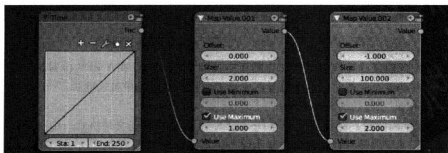


图 10-98 使用多个 Map Value（映射值）结点来完成特殊的数值输出

这里由于 Time（时间）结点的输出值为 0 到 1，那么经过第一个结点的处理，输出值将被叠



加一个乘积2，输出值变为0到2。当连接第二个结点时，前者的输出值会被添加一个值为-1的偏移量，并乘以增益值100，这样最后的输出值范围将为-100至100。

#### 10.4.4.3 Normalize（标准化）结点

最后一种矢量类结点为 Normalize（标准化）结点，它可将所有的输入值都按比例转换至区间0至1。如图10-99所示，为使用 Normalize（标准化）结点转换物体 Z 通道值的结点示例。

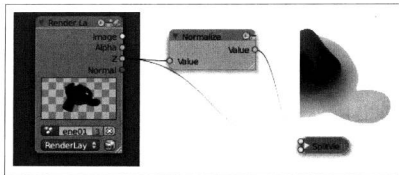


图 10-99 使用 Normalize（标准化）结点转换物体的 Z 通道效果

图10-99中物体的距离离摄像机有10个Blender单位，如果不借助重映射可能无法查看Z通道的数据效果。当连接 Normalize（标准化）结点后，画面范围内的Z通道数据将被按比例缩放至0到1的数值区间，这时就可以很容易地从预览结点中查看Z通道的深度值了。

### 10.4.5 滤镜类结点

#### 10.4.5.1 Filter（滤镜）结点

滤镜类结点能采用不同的处理方式对图像做像素级的更改，以达到一些抽象的特殊化艺术效果。Filter 滤镜结点包含了多种处理方式，如 Soften（模糊）、Sharpen（锐化）、Laplace（拉普拉斯模糊）、Sobel（索伯儿边缘计算）、Prewit（改进的索伯儿边缘计算）、Kirsch（基尔系边缘计算）和 Shadow（外阴影计算）。如图10-100所示，为各种滤镜模式下的处理效果。

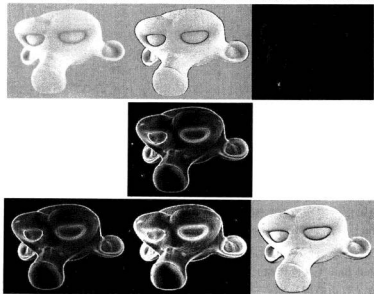


图 10-100 Filter（滤镜）结点下各种模式的计算效果

#### 10.4.5.2 Blur (模糊) 结点

Blur (模糊) 结点用于提供不同级别的模糊效果, 包括 Flat (近似) 模糊、Tent (叠加) 模糊、Quadratic (二次) 模糊、Cubic (立方) 模糊、Gaussian (高斯) 模糊、Fast Gaussian (快速高斯) 模糊、Catrom 模糊和 Mitch (米奇) 模糊。其中, Flat 只是简单地对所有物体都执行近似模糊处理, Tent 则在边缘产生线性的衰减化模糊, Quadratic 和 Catrom 在模糊过程中能保留清晰的边缘, 而 Cubic 和 Mitch 的边缘将显得更平滑。如图 10-101 所示, 为各种模式下的物体模糊效果。

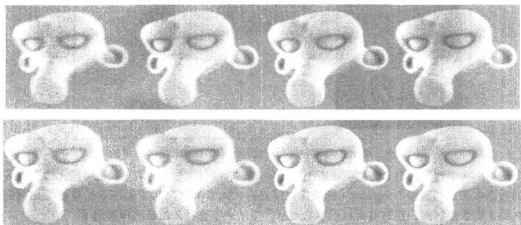


图 10-101 Blur (模糊) 结点下各种模式的效果

#### 10.4.5.3 Directional Blur (方向模糊) 结点

Directional Blur (方向模糊) 结点是一种特殊的矢量方向模糊结点, 如图 10-102 所示, 为 Directional Blur (方向模糊) 的处理效果。

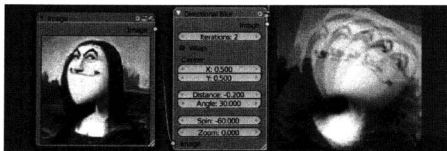


图 10-102 Directional Blur (方向模糊) 结点的处理效果

Iterations (迭代) 参数用于设置图像的迭代过渡数量, 值越大画面的模糊越连贯。当激活 Wrap (包围) 选项后, 模糊计算将保留画面的边缘效果。而 Center 为模糊旋转的中心坐标值, 使用 X 和 Y 来相对定位旋转原点与图像正中间的偏移量。Distance (距离) 值定义了旋转原点和图像原点间的距离, Angle (角度) 则是这两点之间的夹角值。Spin (旋转角度) 值的大小控制了画面本身的旋转角度, 而 Zoom 则是旋转后的画面缩放比例值。

#### 10.4.5.4 Bilateral Blur (双边模糊) 结点

Bilateral Blur (双边模糊) 结点产生的模糊效果会更加柔和和梦幻, 它能保存边缘的一些细节。如图 10-103 所示, 为 Bilateral Blur (双边模糊) 结点的模糊效果。



图 10-103 Bilateral Blur (双边模糊) 结点的处理效果

Iterations (迭代) 参数定义了模糊的深度, Color Sigma (色彩西格玛) 值则用于限制模糊区域的色彩差值, 而 Space Sigma (空间西格玛) 值用于设置模糊半径。Determinator (输入接口) 只有在有接入的时候才起作用, 结点将根据这里的图像像素分布来定义画面的模糊边缘位置。

#### 10.4.5.5 Vector Blur (矢量模糊) 结点

Vector Blur (矢量模糊) 结点常用于制作物体的运动模糊, 为动画带来真实的运动感。在使用矢量计算前, 需要开启渲染层的 Speed (运动) 输出, 这样才能将物体的运动信息导入至结点, 用于计算物体运动的矢量方向。如图 10-104 所示, 为 Vector Blur (矢量模糊) 结点的应用示例。

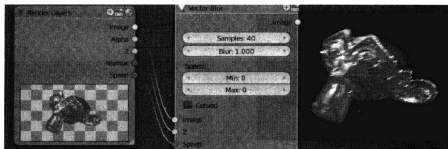


图 10-104 Vector Blur (矢量模糊) 结点的处理效果

Samples (采样率) 决定了矢量模糊的品质, 采样率越高模糊连续性质量越高。同时 Blur (模糊) 参数可用于调节矢量方向上的模糊长度, 值越大矢量方向的模糊范围越长。Speed (速度) 值则用于约束物体在结点计算时的速度极值, 例如当物体运动过快导致像素的运动模糊出现不连续性时, 如果设置了最大速度值, 这些像素点将不会被高速度的矢量运算给完全模糊掉, 而是使用极值速度来限制计算其模糊效果。

#### 10.4.5.6 Dilate/Erode (膨胀) 结点

Dilate/Erode (膨胀) 结点, 可将画面整体膨胀扩大, 并且根据 Distance (距离) 值来计算膨胀后的画面尺寸。如图 10-105 所示, 为使用 Dilate/Erode (膨胀) 结点的处理示例。



图 10-105 Dilate/Erode (膨胀) 结点的处理效果

### 10.4.5.7 Defocus (景深) 结点

DOF (Depth of Field, 景深) 是现实摄影中描述空间中清晰成像距离范围的方法, Defocus (景深) 结点就用于创建镜头中物体的景深效果。现实中的镜头只能够在光聚集在某一固定距离时才会产生清晰成像, 而当远离这点时影像将变得逐渐模糊, 这段距离就是景深距离。景深通常用镜头的光圈值控制, 镜头的光圈值越大, 光圈尺寸将缩小, 景深将得以加深。如图 10-106 左图所示, 为镜头的小光圈结构。反之如果光圈值越小, 光圈尺寸将被放大, 景深感也将被减弱。如图 10-106 右图所示, 为大光圈值时的光圈结构, 而中图为镜头的正常的 aperture 尺寸。



图 10-106 物理镜头的光圈结构

Blender 中的镜头没有光圈的属性, 可以利用 Z 通道来近视地计算场景的深度。如图 10-107 所示, 为利用 Defocus (景深) 结点制作的场景景深效果。



图 10-107 Defocus (景深) 结点制作的场景景深效果

可以在 Bokeh Type 中选择不同的景深成像类型, 其中 Circular 的模糊效果最好, 其次是 Triangular 计算, 而 Octagonal 的质量最差。Angle 角度值为 Bokeh 计算时的边缘衰减角度, 当 Gamma Correction 激活后结点能对聚焦区域的边缘做平滑处理。fStop 参数用于模拟的光圈值, 由于真实镜头中光圈值越小, 光圈越大, 景深的深度效果越浅, 所以当这里设置为 128 时, 镜头的深度将为无穷远, 也就是完全没有模糊感。Max Blur (最大模糊) 值用于限制模糊的半径范围, 当设置为 0 时表示没有限制。Threshold (阈值) 参数控制了聚焦区域和模糊区域的过渡, 通常保持默认即可。Preview (预览) 模式用于激活预览采样, 但是会降低显示质量, 在最后的输出渲染时应关闭这里的 Preview 选项。Use Z-buffer 将开启 Z 通道输入来辅助图像的景深渲染, 如果没有 Z 数据通道则可以关掉这个选项, 结点将使用一个灰度值来代替通道深度。最下面的 Z-Scale 通道缩放参数用于调节 Z 通道的深度比例, 控制景深图像的模糊深度效果。要使用 Defocus (结点) 计算景深, 需要设置镜头的 Limit (极限) 属性, 调节镜头的 Distance 属性, 将光标移动至镜头的焦点位置, 并开启渲染的 Z 通道输出, 如图 10-108 所示。

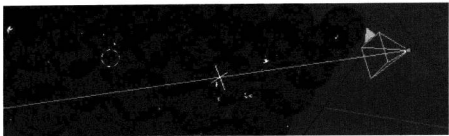


图 10-108 设置摄像机的聚焦位置

#### 10.4.5.8 Glare (眩光) 结点

Glare (眩光) 结点可为图像添加一些特殊的发光效果, 如图 10-109 所示, 使用 Glare (眩光) 结点为车灯添加了特殊的光影效果。



图 10-109 使用 Glare (眩光) 结点添加特殊光影的效果

Glare 提供了多种光影效果, 包括 Streaks、Simple Star、Fog Glow 和 Ghosts。如图 10-110 所示, 分别为这几种模式下的特殊眩光效果。

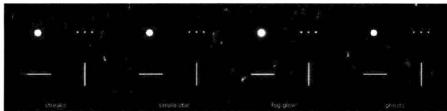


图 10-110 不同模式下的特殊眩光效果

Iterations 用于计算结点的迭代次数, 值越高计算结果的质量越好。Color Modulation (色彩调整) 则可调节光影的光谱效果, 其中 Mix 的值为原图与眩光效果的混合比例值, 当值为 -1 时, 仅输出原图效果, 而当值为 1 时, 输出结果为纯眩光效果, 合理地调节 Mix 值可控制光影在原图上的混合强度。Threshold 阈值参数调节了色彩上的高光偏移量, Streaks 用来控制发射光线的数量, Angle Offset 的值为光线之间的夹角度数, 而 Fade 值可用来调节光线与背景之间的融合度。

#### 10.4.6 转换类结点

合成类结点中的转换类结点包含了与材质类中相同作用的 Color Ramp、RGB to BW、Math 结点, 同时还添加了更多的色彩分离结点, 包括 Separate RGBA、Combine RGBA、Separate HSVA、Combine HSVA、Separate YUVA、Combine YUVA、Separate YCbCrA 和 Combine YCbCrA, 这些色彩

分离结点可将图像色彩按照不同模式分离成不同的色彩通道，用于单独调节或特殊调用。

#### 10.4.6.1 Set Alpha（透明设置）结点

Set Alpha（透明设置）结点，可为一些不支持透明通道的图像，例如 JPEG 格式，添加一个 Alpha（透明）通道。如图 10-111 所示，为 Set Alpha（透明设置）结点的应用示例。

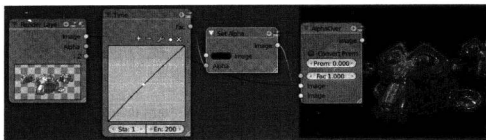


图 10-111 Set Alpha（透明设置）结点的处理效果

Image 输入接口可接图像输入，也可以接选择色彩输入，Alpha 值为画面的透明度。对于图 10-111 中的示例，使用 Time（时间）结点的输出值来控制叠加色彩的透明度，当时间帧运行至 200 帧时，透明度将被设置为 1，这时黑色图层将完全覆盖住渲染图层中的物体，实现了图片渐隐的动画效果。

#### 10.4.6.2 ID Mask（序列遮罩）结点

ID Mask（序列遮罩）结点，可在渲染输出时按照序列编号，输出不同物体的 Alpha（透明）通道。如图 10-112 所示，为 ID Mask（序列遮罩）结点分别输出每个物体的透明通道结点示例。

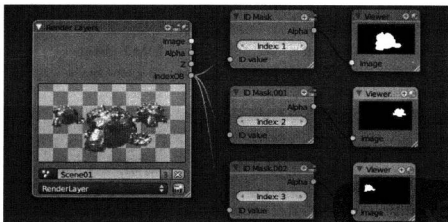


图 10-112 使用 ID Mask（序列遮罩）结点分别输出各物体的透明通道效果

在使用 ID Mask 结点运算透明通道前，需要在属性面板中为每个物体分别指派一个序列 ID，并在渲染输出中开启 Object Index 输出，如图 10-113 所示。

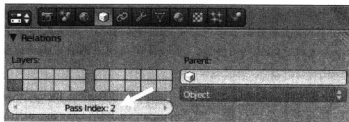


图 10-113 为物体指派序列 ID

### 10.4.6.3 Alpha Convert (透明度转换) 结点

Alpha Convert (透明度转换) 结点, 用于转换物体的 Alpha 类型。在高级合成中, 例如将实拍视频与动画结合时, 普通计算运动模糊的边缘使用的是默认 Key 透明设置, 其色彩边缘的过渡和透明通道的边缘均为线性过渡至色彩本身, 如图 10-114 左图所示。这样的合成效果在物体周围将会产生深色的不自然线条, 降低了画面的可信性, 如图 10-114 中图所示。而一个满意的合成效果应该如图 10-114 右图所示, 其动画物体与背景在边缘的合成处应该是渐变至透明的效果。



图 10-114 不自然的过渡效果

所以利用 Alpha Convert (透明度转换) 结点, 如图 10-115 所示, 可将透明通道的边缘透明模式转换为 Premultiplied 模式, 这样再通过 Set Alpha 结点合成时, 便可将运动物体的边缘自然地融合至背景色彩中了。

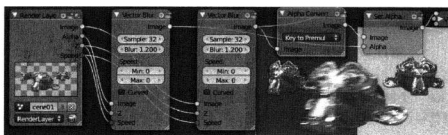


图 10-115 使用 Alpha Convert (透明度转换) 结点来修改透明的计算模式

## 10.4.7 影像类结点

### 10.4.7.1 Difference Key (容差) 结点

Matte (影像) 类结点可用于蓝幕或绿幕合成, 也可用于特效处理等应用。绿幕前的物体可以称为前景色, 绿幕以及需要被过滤掉的色彩被称为背景色。如图 10-116 所示, 为 Difference Key (容差) 结点用于去除绿幕背景的应用示例。

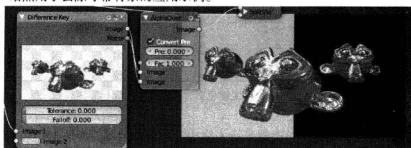


图 10-116 去除绿幕背景的效果

Difference Key (容差) 结点可用于计算图像中的色差, 其中 Tolerance (容差参数) 值用于识别色彩的差值, 通常取值为 0.15, 也就是允许过滤色彩范围在 15% 之内。而 Falloff (衰减) 值则用来控制色彩在差值范围内的衰减强度, 值越高, 离过滤色差越近的像素衰减越快, 而较远的色彩像素的过滤衰减将越慢。具有同样功能的还有 Chrome Key (色度) 结点, 它们的功能相同, 但取色原理和取色方式不同。

#### 10.4.7.2 Distance Key (偏移) 结点

Distance Key (偏移) 结点可根据 Tolerance (容差) 值, 动态地选取图像中与特殊色彩具有相关性的区域。如图 10-117 所示, 为使用 Distance Key (偏移) 结点, 选取图像中特定颜色及其一定 Falloff (衰减) 范围区域内色彩的结点示例。

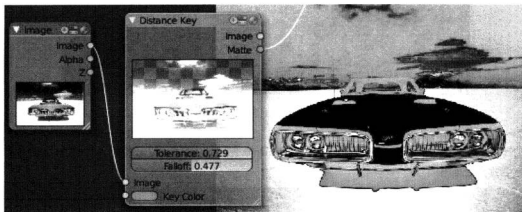


图 10-117 使用 Distance Key (偏移) 结点提取指定色彩区域的效果

#### 10.4.7.3 Color Key (关键色彩) 结点

Color Key (关键色彩) 可用于提取画面中指定的色彩区域。如图 10-118 所示, 为使用 Color Key (关键色彩) 结点, 提取图像中特定颜色的结点示例。其中 Key Color 用于选择需要提取的关键色彩。

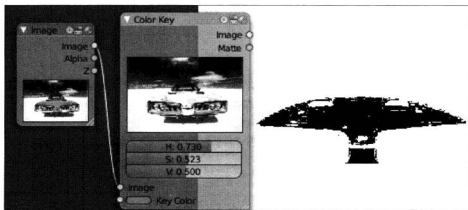


图 10-118 使用 Color Key (关键色彩) 结点提取指定色彩区域的效果

#### 10.4.7.4 Channel Key (通道) 结点

Channel Key (通道) 结点, 可根据通道来完成对色彩的分离。如图 10-119 所示, 为选择 RGB 色彩空间并对 R 红色通道做着色处理的结点示例。





## 10.4.8 变形类结点

### 10.4.8.1 Displace (置换) 结点

Distort (变形) 类结点能提供常用的形变功能, 包括 Translate (位移) 结点、Rotate (旋转) 结点、Scale (缩放) 结点, 以及用于镜像变化的 Flip (反转) 结点、修剪画面的 Crop (裁剪) 结点等。除了对画面的物理修改, 变形类还提供了 3 种特效变形结点。如图 10-122 所示, 为 Displace (置换) 结点的示例效果。

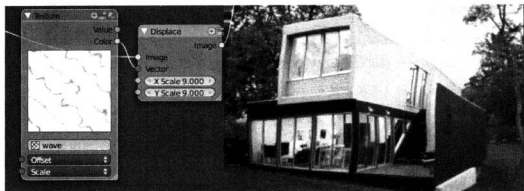


图 10-122 Displace (置换) 结点的处理效果

Displace (置换) 结点可将一个输入值应用至图像的法向显示上, 产生出置换的特殊效果, 例如制作沙漠中的热空气特效。

### 10.4.8.2 Map UV 结点

Map UV 结点可重新定义物体的 UV 贴图, 这时渲染的物体必须拥有一个 UV 图层, 并需要在渲染面板中开启图层的 UV 渲染输出。结点中的 Alpha 选项用于控制新的贴图与原贴图的混合深度, 如图 10-123 所示, 为 Map UV 结点的重贴图示例效果。

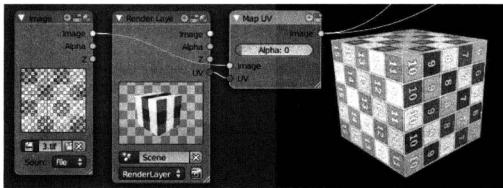


图 10-123 使用 Map UV 结点重贴图的 UV 效果

### 10.4.8.3 Lens Distortion (焦距失真) 结点

最后一种结点为 Lens Distortion (焦距失真) 结点, 这是一种十分强大的物理镜头焦距模拟结点, 它能产生物理镜头在景深边缘位置出现的光影效果。如图 10-124 所示, 为 Lens Distortion (焦距失真) 结点的 Jitter 抖动效果。



图 10-124 使用 Lens Distortion (焦距失真) 结点为画面添加抖动的特效效果

## 10.5 视频序列编辑器

Video Sequence Editor (VSE, 视频序列编辑器) 是 Blender 集成的一款 Nonlinear Editing (非线性编辑) 视频处理工具, 类似于常用的 Adobe Premiere 等专业视频处理软件, 它可以将通过结点工具处理过的渲染序列或图像组合成一个完整的视频输出。非线性编辑的概念就是可以自由地处理视频中任意位置的画面, 而无需像传统的线性视频编辑一样, 通过从头至尾地回来反复记录并编辑视频的片段。如图 10-125 所示, 为 VSE 的编辑面板。

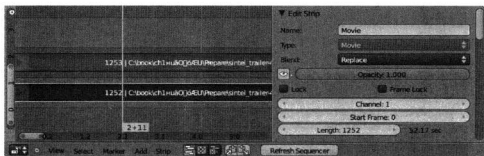


图 10-125 Video Sequence Editor (视频序列编辑器) 的面板

### 10.5.1 序列编辑器界面

单击状态栏上的选项图标, 可让视频编辑器的界面切换至不同的观察和功能窗口。默认的 VSE 为编辑窗口, 其菜单按钮为 。在这个窗口中可以导入视频序列, 编辑并剪辑视频。单击按钮 可切换至视频预览窗口, 如图 10-126 所示, 为输出视频的预览效果。



图 10-126 输出视频的预览效果

### 10.5.1.1 Luma Waveform (照度波)

除了对输出视频的预览，VSE 还可输出详细的色彩信息，例如 Luma Waveform (照度波)。Luma 是非线性视频编辑中的亮度颜色空间分量量度标准，用于表示色彩亮度和色彩对比度。由于人眼对于环境中对比度较高的区域会十分敏感，所以根据 Luma 的信息，可以清晰地了解画面中对人眼的吸引点在哪，并且根据需要来调整图像，以引导观众的视觉导向。

如图 10-127 所示，以《Sintel》剧照为例，画面中以一个深邃的巷道为背景，Sintel 在废墟中搜索。VSE 的 Luma 图将以画面的宽度为 X 横坐标，Luma 亮度值为 Y 纵坐标。从 Luma 图中可以看到，左 1/3 处和右 1/3 处的 Luma 值最高，也就是说这两点是最吸引视觉焦点的位置。转向画面可发现，左 1/3 处刚好为主角 Sintel，而右 1/3 处为被较强光线照射的一面墙体，这两者的亮度对比值最高。所以观众在观看影片时的第一视觉就会被墙体上的大片静态反光吸引，首先感受环境中的气氛，而动态的 Luma 图像将会转移人眼的注意力，所以观众接下来将随着主角的移动，把视觉注意力转向到人物的动作上，观众就这样跟随导演的引导逐渐走入到故事的情节中。因此，Luma 值也成为影片用于控制观众视觉焦点和注意力的重要工具之一。

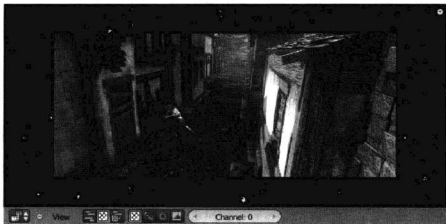


图 10-127 Luma 值控制观众视觉焦点和注意力

同时，Luma 值还包含了画面的色彩信息，从图 10-128 中可以发现，右上角的墙体色彩拥有较高比例亮度的黄色，在查看 Luma 时可以看到，这是因为该区域上的红色值和绿色值比例较

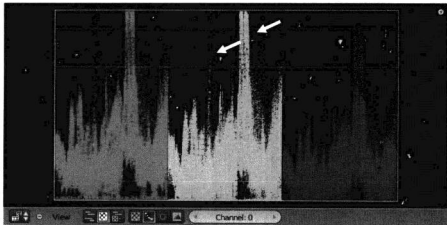


图 10-128 使用 Luma 照度波值来调整图像的视觉导向

高。因此如果希望营造一种特殊的气氛,也可以使用 Luma 来监控对画面色彩的控制和搭配结果。例如,如果要表现一个紧张而危险的场面,可以在画面的不同位置处添加一定的红色,并参考 Luma 来调整色彩在图像中的分布,以达到正确控制画面氛围的目的。

#### 10.5.1.2 Chroma (色彩差值分量)

为了查看更详细的色彩信息,Luma 值常需要配合使用 Chroma (色彩差值分量),这个差值分量是基于 RGB 色彩与 Luma 的差值,用来表示视觉对色彩的感知。事实上各种颜色的鲜艳程度和人眼的感官直觉并不是很一致的,而人眼却可以更容易地区分出色度空间中由于色彩间的差距而造成的光线强弱,所以常使用差值分量来调节画面的色彩分布。

VSE 的 Chroma 查看器将色带分布在一个六边形中,每个边代表不同的色彩分布,当画面中的色调倾向某一个色彩空间时,色彩像素的分布会向对应的六边形区间发散。如图 10-129 所示,为《Sintel》中的一个画面,可以看到画面的主色调偏黄与红,查看对应的 Chroma 图也发现,色彩的分布也集中在黄至红的区间之内。在对影片中同一个场景内不同的画面间执行切换时,可能希望这些画面的整体色彩可以一致,那么合理地使用 Chroma 来查看色彩空间的分布,就能方便而快速地完成每个镜头的色彩调整了。

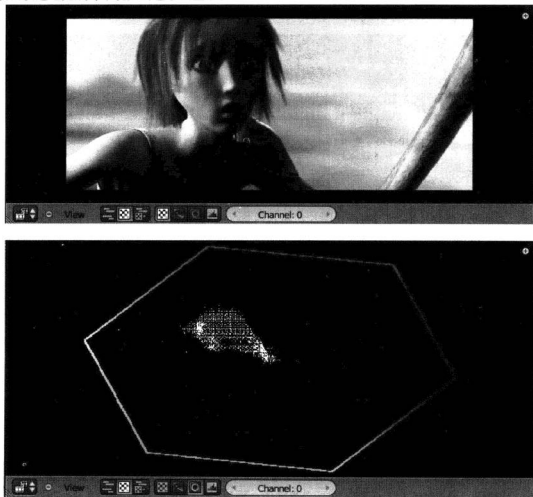


图 10-129 使用 Chroma (色彩差值分量) 来查看色彩的明度空间

### 10.5.1.3 Histogram (直方图)

Histogram (直方图) 则更直接地显示了色彩的像素分布, 场景中的某一个色彩值数量越多, 那么直方图中它的纵坐标值也就越高。如图 10-130 所示, 上图为《Sintel》中一幕沙漠场景, 下图为其色彩直方图分布。



图 10-130 使用 Histogram (直方图) 来查看画面的色彩分布

## 10.5.2 序列编辑

如图 10-131 所示, 在 Add 列表中可以选需要添加的素材类型, VSE 支持 Movie (视频) 输入、Sound (音频) 输入、Image (图片) 输入和 Scene (场景) 的渲染导出, 不同的素材类型将使用不同的颜色来进行区分。可以像选择 Object (物体) 一样单击 RMB 来选择这些素材片段, 然后使用快捷键 G 来移动这些片段至不同的编辑通道。



图 10-131 添加 Strip 素材文件


在选择并添加素材后, 还可以使用 Strip 片段菜单提供的各种编辑工具, 来对视频或图像执行剪辑操作。例如, 可以使用快捷键 K 将一个完整的视频, 按照帧的位置做分段处理, 也可以使用 E 来移动或者拉伸片段的长度, 并用快捷键 M 将多个独立的素材片段合并起来组成一个 Meta 片段组。单击状态栏上的  可在一个界面中, 同时预览编辑窗口和剪辑结果, 如图 10-132 所示。



图 10-132 使用 Strip 工具来编辑视频素材

### 10.5.3 特效

VSE 提供了一些常用的视频剪辑特效，当然也可以添加自定义的外置特效。VSE 的大部分特效依赖于一个或多个基础素材的属性，所以需要配合使用快捷键 B 来框选多个素材。

首先添加两个图片素材，分别放置在通道 0 和通道 1，接着使用快捷键 B 框选这两个素材，并添加一个 Add（叠加）特效片段，如图 10-133 所示。



图 10-133 Add（叠加）特效的效果

**Add:**（叠加）特效可将两个素材中的色彩混合起来，其色彩叠加原理和结点的 Mix（混合）模式类似。

**Subtract:** 相减特效，可对图层间的色彩值做减法计算，效果如图 10-134 所示。

**Alpha Over:** 透明通道叠加特效，可将上层图层按照透明通道叠加至下一个图层上，效果如图 10-135 所示。类似功能的特效还包括 Alpha Under 和 Over Drop。

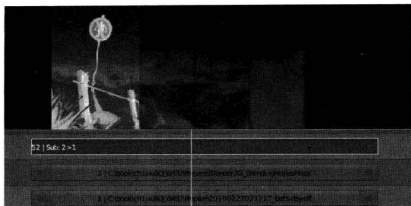


图 10-134 Subtract（相减）特效的效果

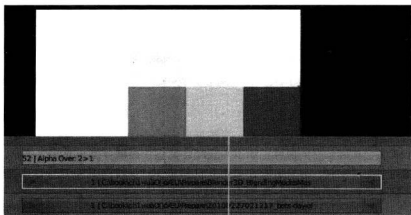


图 10-135 Alpha Over（透明通道叠加）特效的效果

**Cross:** 过渡特效，常用于制作两片段之间的渐变过渡特效，如图 10-136 所示。类似功能的特效还有 Gamma Cross 特效。

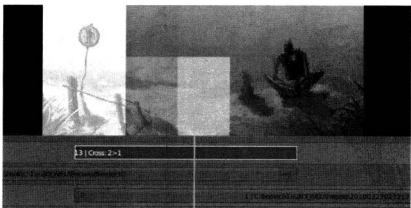


图 10-136 Cross（过渡）特效的效果

**Glow:** 发光特效，可为画面添加高光效果，如图 10-137 所示，单击 N 可调出特效的属性面板，用于控制 Glow（发光强度）和模糊强度。其中，Clamp 用于调节色彩的光辉，值越大光照



越强，而 Blur Distance 则用来控制模糊的尺寸。



图 10-137 Glow（发光）特效的效果

上面简单介绍了 Effect 中的几个最常用的特效，剩下的特效就留给读者自行练习和实践。

#### 10.5.4 序列输出

在完成视频编辑后，就可以渲染并输出视频了。视频的输出版可在渲染面板中设置，如图 10-138 所示，包括 Resolution（画面的尺寸）、Frame Range（输出的时间轴长度）以及 Frame Rate（帧率）等。对于常用的视频输出设置，请参考第 7 章中的渲染功能说明。

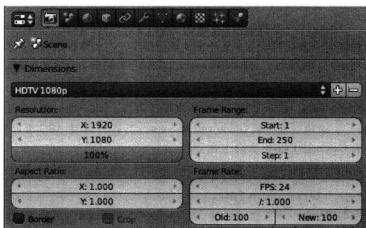


图 10-138 设置文件的输出格式

接下来还需要在 Post Processing 面板中开启 Sequencer（序列）渲染功能，如图 10-139 所示，这样才能启动 VSE 的视频输出。最后单击 Render（渲染）面板中的 Animation（动画）渲染，或者使用快捷键 Ctrl + F12，即可执行 VSE 的序列渲染输出了。

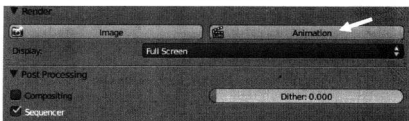


图 10-139 激活 VSE 渲染输出

## 第 11 章

# Python脚本扩展

Blender 集成了 Python 为自身的脚本语言，它使用更直接的方式去控制场景中的数据，强化现有的功能模块，提供更多的自定义选项，并完成特殊操作动作（如批处理等）。

在 Blender 中有两种方法调用 Python 完成功能扩展：第一种方法是直接在内部的脚本编辑器中运行脚本，然后在编译出的界面中执行功能调用；第二种方法是编写调用链接脚本，将外部扩展程序链接至 Blender，使系统可以将数据导出至外部程序中进一步的处理。

## 11.1 Python 脚本基础

Blender 不是一个静态的封闭式程序，它不需要经过再编译，就可以直接根据 Python 脚本实时修改软件界面和功能。Blender 现在已拥有数百个不同功能的脚本，由全球各地的用户自行编写并上传共享，可以登录官方网站的下载页面<sup>①</sup>，或者 Blender 艺术家论坛的脚本专区<sup>②</sup>下载使用这些脚本。

### 11.1.1 Python 简介

Python 是一种面向对象、直译式的计算机程序设计语言，同时它也是一种功能强大而且完善的通用型脚本语言。它由荷兰程序员 Guido van Rossum，于 1989 年在 ABC 语言基础上开发出来。如图 11-1 所示，为 Guido 在 Python 大会上发言。

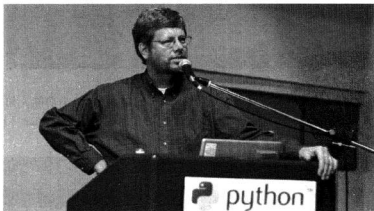


图 11-1 Python 的创始人 Guido van Rossum

① 网站地址：<http://wiki.blender.org/index.php/Extensions:2.5/Py/Scripts>。

② 网站地址：<http://blenderartists.org/forum/forumdisplay.php?f=11>。

作为一种高级的动态编程语言，Python 对于 C 和其他语言有着良好的支持特性，它同时还拥有十分强大的标准库，能由用户做自定义扩充，使其可以和由其他语言编写的程序进行集成和封装。因此，至今很多人还把 Python 称为 Glue Language（胶水语言）。

最新版本的 Blender 集成了 Python3.0，简称为 Py3k。相比早期的 Python 版本，它主要的改进包括部分函数和语法的更新，以及对一些字符串的重定义，最重要的是它取消了向下兼容的功能，这就使得在早期 Blender 版本下编写的脚本程序，可能无法在最新版本的 Blender 环境中运行使用。

### 11.1.2 编辑与编译

Text Editor（文本编辑器）是在 Blender 下编写 Python 脚本的主要工具，可以单击窗口选项将工具界面切换至 Scripting（脚本布局），在弹出的 Text Editor（文本编辑器）中即可编写程序代码，如图 11-2 所示。

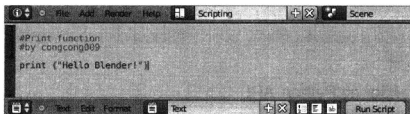


图 11-2 使用 Text Editor（文本编辑器）来编写 Python 脚本

在完成脚本的编写后，可以单击状态栏上的 Run Script（运行脚本）按钮，或者使用快捷键 Alt + P。运行的结果将显示在控制台上，可以切换至控制台查看，打印结果如图 11-3 所示。

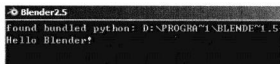


图 11-3 使用控制台查看脚本运行信息

也可以使用一些已集成的模板，来快速地完成脚本编辑。如图 11-4 所示，为 Text Editor 中提供的部分应用模板。

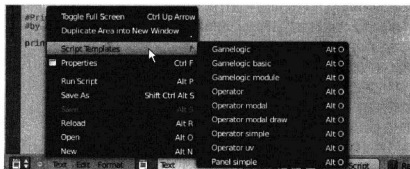


图 11-4 使用现成的 Script Templates（脚本模板）

除了能使用 Text Editor 来编辑脚本，还可以在 Console（控制台）窗口中，直接输入命令行来快速执行或调用功能，如图 11-5 所示。

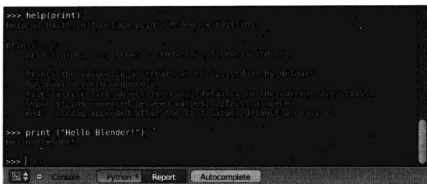


图 11-5 在 Console（控制台）下直接运行 Python 命令

在任意窗口中，使用快捷键 Shift + F4 均可切换至 Console（控制台）窗口。单击状态栏上的 Autocomplete 按钮可自动编译输入的语句。也可以使用快捷键 Ctrl + Space 快速激活 Autocomplete（自动编译）功能。

## 11.2 Blender 的 Python API

### 11.2.1 API 基础

Blender 的 Python API（Application Programming Interface，应用程序编程接口）是一个预先定义好的函数库，目的是使 Blender 和脚本开发人员可以通过直接使用这些函数即可完成对数据的调用，而无需访问源代码去了解内部函数方法的执行细节。Blender Python API（BPY）是 Blender 使用 Python 与系统执行数据交换和功能调用的接口模块，具体的访问方法依赖于模块中定义的调用接口类型。本章中涉及的 API 接口都为 Blender 的 BPY 模块应用，这个模块应区别于 Blender Game Engine（BGE）中应用的 API 接口，虽然前者的一些功能还可以被应用在 BGE 的 Context（正文）中。本书并不专注于程序基础知识的讲解和编写范例，关于 Python 编程基础，还是请读者去查看程序自带的 API 文档。当能够熟悉 90% 以上的 API 文档描述时，就已经能够成为一位出色的 Blender Python 脚本设计师了！

通过 BPY 的 API 函数，可以使用脚本执行以下功能：

- ☐ 代替界面操作去完成对物体的修改，例如修改网格属性或添加修改器。
- ☐ 自定义系统的相关配置，例如重设快捷键或修改主题的色彩等。
- ☐ 自定义工具的参数配置，例如自定义雕刻笔刷的参数。
- ☐ 自定义用户界面，例如修改面板的外观和按钮的排列效果。
- ☐ 创建新的工具，例如 Surface Sketching（表面绘制）工具。
- ☐ 创建交互式工具，例如游戏的逻辑脚本。
- ☐ 创建与外置渲染器的接口调用，例如配置 Vray 等外置渲染器。
- ☐ 重定义 Blender 的数据结构。

Blender 的界面中，在每个按钮上单击 RMB，都可以得到相关功能的 API 说明文档，以便于在脚本编写中找到对应功能的方法描述。如图 11-6 所示，在渲染按钮上单击 View Docs 选项，即可打开与 Render 方法相关的 API 文档。

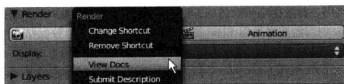


图 11-6 查看功能的相关文档

### 11.2.2 事件查看

Python 是一种事件型语言，因此它的响应反馈必须建立在已注册的事件基础上。在 Console（控制台）中，当输入如下语句：

```
> > bpy.
```

按回车键后会发现控制台报错，这是因为还没有对 bpy 调用任何事件。如果在输入后单击 Autocomplete 按钮，或者使用快捷键 Ctrl + Space，控制台将自动显示当前语句的配置信息，返回值为 bpy. 模块类中的注册事件，如图 11-7 所示。



图 11-7 使用 Autocomplete 来查看已注册的配置信息

这时继续输入如下语句：

```
> > bpy.context
```

再次单击 Autocomplete，就能在控制台中查看到当前 Context（正文）中的全部注册事件了，如图 11-8 所示，这些事件都可以继续通过 Python 的 API 来直接调用。因此在程序设计中，合理地使用 Autocomplete 可以有助于更快地掌握库结构的信息，并熟悉 API 的各种接口细节。



图 11-8 在正文中的全部注册事件

### 11.2.3 数据访问

前面曾提到，任何从界面对物体的操作，都可以使用 Python 脚本来替代，因此了解 Blender 中的 Data（数据）结构就显得十分重要。只有掌握了数据的结构和访问原理，才能灵活地使用相关函数，完成对物体对象的访问和修改，以及对动画属性等参数的调用。

在 Python 中，任何物体都是一个 Object（对象），这和 Blender 中的 Object（物体）含义很容易混淆。Blender 中的 Object 是指场景中的物体，例如网格和灯光等，它们在界面中与用户实现直接的 Interaction（动作交互）。Python 的 Object 则是在系统内部，承载了 Blender 中 Object（物体）的所有 Module（模块）数据信息，包括数据类型和方法接口等，所以又称之为 Data Block（数据块）。当在视图中新建一个 Object（物体）时，系统会在数据树中自动生成一个对应的 Object（对象）数据块，前者的调用对象是用户，而后的调用对象则是 Blender 的数据管理系统。

Data API 能提供用户对数据结构的访问方法,例如在外置渲染器中可以通过渲染 API 来读取场景的数据。Blender 内部的数据是以 Class 类的方式存储于 RNA 型数据结构中,可以在 Library Data (数据库) 中很容易地查找到与其对应关联的数据属性。同时所有的数据访问又使用了一个继承于 Main (主类) 的实例 instance bpy.data, 使用它的注册方法来搜索并访问 Library Data (数据库) 中的数据模块。

可以在大纲视图中查找数据树上对应的物体属性,如图 11-9 所示。大纲中的属性结构按照继承关系进行排列,Main (主类) 下显示了所有的属性类,每个属性都可以使用继承于 Main 的实例来执行调用和修改。

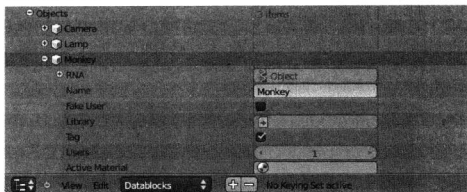


图 11-9 使用大纲查看类的属性

在控制台中输入如下代码,用于查看场景中已有的物体。

```
>>> for object in bpy.data.scenes['Scene'].objects:print(object.name)
```

由于场景中只有一个网格物体、一个镜头和一盏灯,所以最后的输出结果如图 11-10 所示。

```
>>> for object in bpy.data.scenes['Scene'].objects: print(object.name)
...
Lamp
Camera
Monkey
```

图 11-10 打印查看场景中的物体

除了物体名, Object (物体) 的所有属性也可以被打印查看并执行修改。使用如下语句,用来打印物体的坐标位置属性。

```
>>> for object in bpy.data.scenes['Scene'].objects:print(object.name + "'s location is " + str(object.location))
```

如图 11-11 所示,所有物体的坐标数据都将以矢量的方式打印在控制台上。

```
>>> for object in bpy.data.scenes['Scene'].objects: print(object.name + "'s location is " + str(object.location))
...
Lamp's location is Vector((-0.0, 0.0, 0.0))
Cam's location is Vector((-0.0, 0.0, 0.0))
Monk's location is Vector((-0.0, 0.0, 0.0))
```

图 11-11 打印查看物体的属性

#### 11.2.4 操作符

在 Blender 中需要将完成的 Script (脚本) 注册为 Operator (操作符), 这样才能被允许作为函数方法用来调用数据中的属性参数。这些 Operator 可以在控制台中执行, 也可被集成至界面中由按钮操作来完成。Blender 的 Python API 提供了大量已注册的现成操作符供调用, 接下来就简单讲解 Operator (操作符) 的使用方法。

### 11.2.4.1 调用已注册的操作符

bpy.ops 包含了所有已注册的 Operator，Blender 界面上所有按钮对应的调用方法都被保存在这个类下。当把鼠标移至界面地按钮上时，其调用的 Operator 路径将被完整地显示出来，如图 11-12 所示。



图 11-12 查看按钮的 Operator 路径

Operator 可以直接在控制台上执行，也可以使用脚本来完成对功能的调用。首先在 API 文档中，查看对 Add Monkey（添加大圣物体）的 Operator 说明。

```
bpy.ops.mesh.primitive_monkey_add(view_align=False,
enter_editmode=False,
location=(0.0,0.0,0.0),
rotation=(0.0,0.0,0.0),
layers=(False,False,False,False,False,False,False,False,False,False,False,False,False,False,False,False,False,False,False,False,False,False,False,False))
```

其中的参数说明如下：

- ☐ view\_align (boolean, (optional))：是否将新建物体约束至某一个视角（可选参数）。
- ☐ enter\_editmode (boolean, (optional))：是否立即进入新建物体的编辑模式（可选参数）。
- ☐ location (float array of 3 items in  $[-inf, inf]$ , (optional))：新建物体的位置坐标（可选参数）。
- ☐ rotation (float array of 3 items in  $[-inf, inf]$ , (optional))：新建物体的转动参数（可选参数）。
- ☐ layers (boolean array of 20 items, (optional))：新建物体所存放的图层编号（可选参数）。

了解了操作符的具体参数，就可以使用命令行来新建一个大圣物体了。首先在控制台中执行以下语句：

```
>>> layer=[False]*20
>>> layer[0]=True
>>> for Num in range(0,15,3):
bpy.ops.mesh.primitive_monkey_add(location=(Num,0,0),layers=layer)
```

这里希望将新建物体放置在第一层，所以在第一行和第二行代码里将全部图层变量都置为空，仅开启第一个图层，并为其设置属性值为 True。接下来调用 bpy.ops 中的新建物体操作符，使用一个 For 循环来重复创建网格物体，单击回车命令后，控制台的运行结果如图 11-13 上图所示。

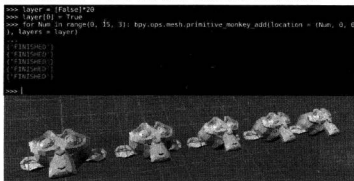


图 11-13 使用命令行直接调用操作符来创建物体

在获得运行成功的信息后，转向 3D 视图窗口，就可以看到 5 个大圣物体按照间隔为 3 个 Blend 单位的顺序，整齐地排列在 X 轴上，创建效果如图 11-13 下图所示。

如果直接调用无参数的 Operator（操作符），系统将为每一个参数值都使用默认设置，也就是说以下两种调用方式将产生同样的效果：

```
>>> bpy.ops.mesh.primitive_monkey_add()

>>> bpy.ops.mesh.primitive_monkey_add(view_align=False, enter_editmode=False, location=(0,0,0), rotation=(0,0,0), layers=(True, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False))
```

它们的运行结果一样，返回值如图 11-14 所示。



```
>>> bpy.ops.mesh.primitive_monkey_add()
{'FINISHED'}

>>> bpy.ops.mesh.primitive_monkey_add(view_align=False, enter_editmode=False, location=(0,0,0), rotation=(0,0,0), layers=(True, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False))
{'FINISHED'}

>>>
```

图 11-14 无参操作符的调用结果

虽然无参的操作符执行更快速简单，但是在脚本设计中，如果不清楚 Blender 对无参操作符的默认参数设置，那么请尽量不要使用类似的调用，因为这样可能会造成不可预料的执行结果。

#### 11.2.4.2 调用自定义注册的操作符

在 Blender 中还可添加并注册自定义的 Operator（操作符）。首先切换至 Text Editor（文本编辑器）中，添加如下代码：

```
import bpy#导入 bpy 库

class HelloBlender(bpy.types.Operator):#通过继承来创建新的操作符
    bl_idname="object.hello_blender"#定义操作符的 id 变量
    bl_label="Hello Blender"

    def execute(self,context):#重写操作符的执行部分
        self.report({'WARNING'}, "Hello Blender!")
        return {'RUNNING_MODAL'}

if __name__ == "__main__":
    bpy.ops.object.hello_blender()#执行调用
```

单击状态栏上的 Run Script 按钮运行脚本，或者使用快捷键 Alt + P，然后等待编译的结束。接着在视图窗口中，单击快捷键 Space 调出快速查找功能，输入 hello 即可找到刚才新建并注册的操作符了，如图 11-15 所示。



图 11-15 查找已注册的操作符

单击运行后，系统将在控制台中打印出警告信息，如图 11-16 所示。



图 11-16 显示运行的结果



由于这个操作符也被成功注册进了 bpy.ops 模块，所以还可以直接在控制台中执行调用。最后的结果如图 11-17 所示。

```
>>> bpy.ops.object.hello_blender()
WARNING: Hello Blender!
{"WARNING": "ADDA!"}
>>>
```

图 11-17 在控制台执行自定义操作符

如果希望在启动时让系统自动注册自定义添加的操作符，可以打开 scripts \ templates \ operator.py 文件，参照其中的模板将自定义的 Operator 注册到 Blender 中即可，这样就可以直接从系统中调用自定义的方法，而无需重复注册了。

### 11.2.5 正文

当完成对操作符的调用，并生成了新的 Object（物体）后，就可以对物体的属性做进一步修改和调整了。对物体的属性操作，实际上就是进入系统的 Context（正文），通过调用正文完成对物体属性的调整。

#### 11.2.5.1 Context（正文）原理

Context（正文）被设计为 Blender 数据管理中的一个全局变量集合，它包含了正被选择激活的物体信息，同时还记录了当前的物体编辑模式，以及参数面板排列和 Blender 窗口类型等信息。如图 11-18 所示，为一个 Context（正文）的结构框图。

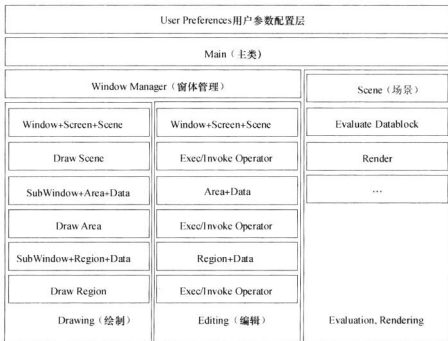


图 11-18 Context（正文）的结构框图

熟悉 Context 的结构可有助于更轻松地掌握当前各可调用变量的控制方法。首先来了解一下 Context 都包含哪些数据类型以及容器的子类。

从结构图中可以看到，Main（主类）是当前的 Blend 工程类，它拥有一个完整的 Data Blocks

数据块，存储了当前工程中的所有属性数据，并可以与其他 Blend 工程文件中的 Main 类建立链接，实现数据交换和访问调用。每一个工程的 Main 类配置，都基于当前用户所建立的一个全局 User Preferences 属性类，因此 User Preferences 类是 Context（正文）的一个外部配置层，所有的代码方法均具有对 User Preferences（属性）类的访问权限。

接下来是基于 Main 的 DataBlocks（数据块），它对 Context（正文）的访问接口被划分为两类：第一类为 Drawing（界面绘制）和 Editing（操作编辑）类，统称为 Window（窗口）管理访问；第二类为内部的渲染调用和数据块评估访问。由 Drawing（界面绘制）和 Operator（操作符）组成的 Window（窗口）级工具，使得通过面板上的 Operator 按钮对数据属性执行修改操作，例如界面代码将通过 OpenGL 渲染粒子面板的外观以及面板上的按钮，而按钮上对应调用的编辑操作符则可以在视图中改变网格的物理属性。对于没有窗口界面的功能，它们将在内部访问工程数据，Context（正文）提供了 Scene（场景）数据来存放这些方法和运行环境。例如，当在物体上添加修改器或约束时，这些修改器内部的执行只运行在窗口的后台，用户无法查看修改器对物体属性的操作过程，系统只返回给用户一个处理的结果。

#### 11.2.5.2 Context 访问流程

Context 本身是只读的，对属性的数据块修改实际上是访问 Context 内部与数据建立起来的调用接口，这可以通过 API 中的 Function（方法）来实现，用于连接数据接口并修改目标类的属性。例如当输入 `bpy.context.object = obj` 时就会产生错误，而使用 `bpy.context.scene.objects.active = obj` 才可以成功转换当前激活的目标物体。这是因为对于 Context 内物体属性的设置并不是直接进入 Context 来完成，而是需要通过 Context 来中转所提出的数据修改操作命令。如图 11-19 所示，为 Context 内的数据访问工作流程。

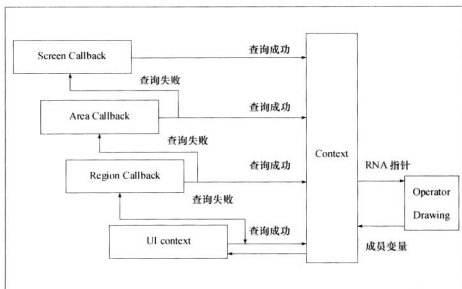


图 11-19 Context 的数据访问工作流程

根据图 11-19 的工作流程，当用户从 UI 界面或在命令行下使用操作符来完成一个指令时，这个指令将被使用 RNA 结构的数据指针收集并提交给 Context。Context 使用了一种 Callback（回调返回）机制，当 Context 接收到请求后就会在 Data Blocks（数据块）中进行遍历查找，一旦检测到了申请调用的目标数据接口，它会直接将这个数据块的 RNA 指针返回给 Context，再由 Context 将结果发送给请求者。

当然，在查找的过程中，Blender 为数据遍历设置了目标优先级，UI 界面结构由于与操作符的结构关系最近，因此它将最先接收到 Context 的查找申请返回值，其次是界面内的 Region（区域）模块，以此逐级传递。如果 Context 没有查找到请求所需要的数据，它将返回一个 NULL 空指针给请求者，并在控制台抛出对应的 Exception（异常）。

### 11.2.5.3 RNA 数据结构

前面提到了数据块的 RNA 结构，RNA 数据结构是从 2.5 版本之后最新引入的一种数据存储结构，它提供了更稳定的动画数据模块，以及更灵活的数据读取方式。RNA 结构是一种自动机理算法，它能像生物 RNA 一样自动创建数据接口的指针，并能快速地读取相邻的数据块，完成在相邻数据间的自动连接。最重要的是，这种数据结构使“将一切物体属性都置入到可实现动画的模块调用中”变为了可能。有幸在 Google 的开发帮助下，Blender 成为了第一款成功部署并使用这种数据结构的软件。如图 11-20 所示，为一个 RNA 数据结构的标准组成框图。



图 11-20 RNA 数据结构的标准组成框图

在 Blender 中任何属性都是由众多 RNA 数据模块构成，这和生物中 RNA 与生物体的组成关系类似。在创建一个 RNA 数据块时，系统将定义一个 identifier（标识符）和用于人工识别的 name（名称）变量，接着系统将会自动遍历结构树，查找具有相同标识符却不同类型的 RNA 数据块，并与其建立自动的关联关系。而相比较早版本的 Blender 存储模式，为一种直接调用 bpy.data 数据树的模式，这种结构的最大劣势，就是它不能像 RNA 一样自由地去访问其他工程 Context 中的全局数据。

### 11.2.5.4 访问 Context（正文）

接下来就试着来访问 Context（正文）中的内容。首先在窗口中选择一个网格物体，输入如下代码：

```
> > > bpy.context.mode
```

按回车键后输出返回结果，控制台将打印当前物体的操作模式，如图 11-21 所示。

```
>>> bpy.context.mode
'edit'
>>> bpy.context.mode
```

图 11-21 查看物体的操作模式

接下来继续输入如下语句：

```
> > > bpy.context.active_object
#激活当前选择的物体数据
> > > bpy.context.object.location.x = 1
#将物体的 x 坐标设置为 1
> > > bpy.context.object.location.x += 0.5
#将物体的 x 坐标加 0.5
> > > bpy.context.object.location = [1,2,3]
#设置物体的坐标为[1,2,3]
> > > bpy.context.object.location.xyz = [3,2,1]
#设置物体的坐标为[3,2,1]
> > > print(bpy.context.object.location)
#打印当前的物体坐标数据
```

在输入数据的同时，观察场景中物体的变化，可以发现，通过 API 访问 Context 不仅可以读取数据信息，还可以很轻松地完成对物体属性的修改操作。

## 11.2.6 类型

Blender 中的 Python API 可以被划分为 3 种 Type（类型）。

### 11.2.6.1 Native（原生）类型

Blender 的控制系统本身就是一款十分强大的 Python 语言 IDE 工具，Native（原生）类型的数据均从 Python 的标准库中导入，包括基础的数据类型定义，例如 Float（浮点数）和 String（字符串）等，常用于数组和数据类型的定义。由于本章并不着重讲解如何完成脚本程序设计，所以对于这些数据类型的定义等基础知识，请读者参考与 Python 编程基础相关的书籍。

先简单查看一下 Blender 中对字符串的操作。首先在控制台中输入以下代码：

```
> > > x = 'hello'
> > > y = 'Blender!'
> > > x + ' ' + y
```

按回车键后查看返回值，如图 11-22 所示。可以看到，这是 Python 标准库中的字符串操作效果。关于这类变量的用法请读者参考 Python 的相关文档，也可以在输入字符后单击 Ctrl + Space 快速查看 Blender 集成的方法说明。



```
>>> x = 'hello'
>>> y = 'Blender!'
>>> x + ' ' + y
'hello Blender!'
>>> |
```

图 11-22 标准字符串运算

### 11.2.6.2 Internal（内置）类型

Internal（内置）类型，为 Blender 的自定义数据类型，包括 mesh（网格）和 bones（骨头）等类型，常用于定义场景中 Object（物体）的属性和结构等。

在控制台中输入如下代码：

```
> > > bpy.types
```

单击快捷键 Ctrl + Space，系统将自动列出当前 Context 中的可调用的内置数据类型，如图 11-23 所示。

```
>>> bpy.types
<Bpy_types object at 0x00025260>

>>> bpy.types.
ACTION_OT_clone ACTION_OT editmeshedit ACTION_OT copy ACTION_OT delete ACTION
_OT duplicate ACTION_OT extrusion_type ACTION_OT frame_jump ACTION_OT page
_in_type ACTION_OT interpolation_type ACTION_OT keyframe_insert ACTION_OT keyf
```

图 11-23 查看 Context 中可调用的内置数据类型

### 11.2.6.3 Mathutils（数学）类型

最后一种为 Mathutils（数学）类型，主要包括了物体的属性数据，例如 object.location 物体的坐标、scene.cursor\_location 光标的坐标等矢量属性。

在控制台中输入如下代码：

```
>>> bpy.context.object.location
>>> bpy.context.object.location.z += 2.0
>>> bpy.context.object.location
```

返回结果如图 11-24 所示。这一类的的数据会以矩阵等数组的形式存储起来，用于表示物体属性的空间特性。

```
>>> bpy.context.object.location
Vector(0.0, 0.0, 0.0)

>>> bpy.context.object.location.z += 2.0
>>> bpy.context.object.location
Vector(0.0, 0.0, 2.0)

>>>
```

图 11-24 查看物体的空间属性

## 11.2.7 集成

Blender 的所有界面均使用 Python API 注册到 Context 环境中，它的原理和注册 Operator（操作符）类似。除了集成界面和操作符，Python 还可用于集成外置的渲染器，因此掌握 Python API 的集成方法，可有助于重建 Blender 中的所有元素，实现对 Blender 的二次开发。

### 11.2.7.1 Panel（面板）集成

和自定义 Operator（操作符）的注册方式一样，面板也以类的方式注册在 Context 环境中。首先打开 Text Editor（文本编辑器），在其中输入如下代码：

```
import bpy

class OBJECT_PT_hello(bpy.types.Panel):
    #从bpy.types.Panel继承面板类
    bl_label = "My new panel"#定义标题
    bl_space_type = "PROPERTIES"#定义面板类型
    bl_region_type = "WINDOW"#定义窗口类型
    bl_context = "object"#定义窗口位置

    def draw(self,context):#定义绘制方法
        layout = self.layout

        ob=context.object#定义物体
        type=ob.type.capitalize()

        row=layout.row()
```

```

row.label(text="Hello Blender User!",icon='WORLD_DATA')
col=layout.column()

row=col.row()
row.label(text="The currently selected object is:"+ob.name)#打印物体名称
row=col.row()

if type == 'Mesh':#判断流程
    row.label(text="It is a mesh containing "+str(len(ob.data.vertices))+" vertices.")#访问网格属性
else:
    row.label(text="It is a "+type+".")

row=layout.row()
row.alignment='RIGHT'#控制排版
row.label(text="The end")

```

编写完成后单击 Alt + P 运行脚本。接着在场景中选择大圣物体，并转至工具栏中的 Object（物体）属性面板，就可以看到刚才新建的菜单已经被自动编译了出来，并实时地显示了网格物体的相关信息，如图 11-25 所示。

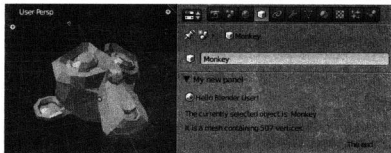


图 11-25 建立自定义的集成界面

### 11.2.7.2 Button 按钮集成

接下来，试着在面板中加入 Button（按钮），并实现按钮在界面上的交互效果。在 Text Editor（文本编辑器）输入如下代码：

```

import bpy
import os

class OBJECT_Panel(bpy.types.Panel):#界面的绘制类
    bl_label="Hello Button"
    bl_space_type="PROPERTIES"
    bl_region_type="WINDOW"
    bl_context="object"

    def draw_header(self,context):#绘制界面的标题栏
        layout=self.layout
        layout.label(text="",icon="PHYSICS")#面板图标

    def draw(self,context):#定义面板的内容
        layout=self.layout

```

```

row = layout.row()
split = row.split(percentage=0.5) #平分界面布局
colL = split.column()
colR = split.column()

colL.operator("pipo",text="Push Me") #定义 Push 按钮
colR.operator("pipa",text="Pull Me") #定义 Pull 按钮

class OBJECT_OT_Push(bpy.types.Operator): #定义操作符
    bl_label = "Push operator"
    bl_idname = "pipo"
    bl_description = "Push Function" #添加注释

    def invoke(self,context,event):
        print("Push Button Pressed") #后台打印
        self.report("INFO","Pushing") #信息栏打印
        return("FINISHED")

class OBJECT_OT_Pull(bpy.types.Operator): #定义操作符
    bl_label = "Pull operator"
    bl_idname = "pipa"
    bl_description = "Pull Function" #添加注释

    def invoke(self,context,event):
        print("Pull Button Pressed")
        self.report("INFO","Pulling") #信息栏打印
        return("FINISHED")

```

单击 Alt + P 运行脚本, 就可以在 Object 面板中找到刚才添加的 Hello Button 面板了。当单击任意按钮时, Context (正文) 中的注册 Operator (操作符) 将自动运行对应的 invoke () 方法, 并在标题栏上打印出相关信息, 效果如图 11-26 所示。

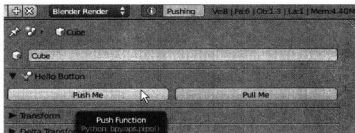


图 11-26 面板上集成的按钮效果

同时系统在控制台, 也将打印程序每次调用操作符后的返回值, 如图 11-27 所示。



图 11-27 后台的打印输出

### 11.2.7.3 Operator（操作符）集成

最后将在按钮中加入对 Context 中数据的访问，以实现通过按钮上集成的 Operator（操作符）完成对物体的创建操作。打开 Text Editor（文本编辑器），输入如下代码：

```
import bpy
import math
import mathutils#导入需要调用的数学库

class OBJECT_PT_Panel(bpy.types.Panel):
    bl_label = "Add Tetrahedron"
    bl_space_type = "PROPERTIES"
    bl_region_type = "WINDOW"
    bl_context = "object"

    def draw_header(self, context):#定义面板标题
        layout = self.layout
        layout.label(text="", icon="PHYSICS")

    def draw(self, context):#定义面板内容
        row = self.layout.column(align=True)
        row.operator("mesh.add_tetrahedron", text="Add Tetrahedron")#为按钮定义操作符以及对应的
```

注册路径

```
class MakeTetrahedron(bpy.types.Operator):
    bl_idname = "mesh.add_tetrahedron"#定义操作符的注册路径
    bl_label = "Add Tetrahedron"
    bl_options = {'REGISTER', 'UNDO'}

    def invoke(self, context, event):#将四边体的顶点坐标分别存进 list
        Vertices = \
        [
            mathutils.Vector((0, -1 / math.sqrt(3), 0)),
            mathutils.Vector((0.5, 1 / (2 * math.sqrt(3)), 0)),
            mathutils.Vector((-0.5, 1 / (2 * math.sqrt(3)), 0)),
            mathutils.Vector((0, 0, math.sqrt(2 / 3))),
        ]
        NewMesh = bpy.data.meshes.new("Tetrahedron")#创建数据块
        NewMesh.from_pydata \
        (
            Vertices,
            [],
            [[0, 1, 2], [0, 1, 3], [1, 2, 3], [2, 0, 3]]
        )#分别输入点、边、面坐标的 list 数据,将多边形网格缝合起来
        NewMesh.update()#更新网格数据块
        NewObj = bpy.data.objects.new("Tetrahedron", NewMesh)
        context.scene.objects.link(NewObj)#将其链接至场景正文中
    return {"FINISHED"}#
```

单击 Alt + P 运行脚本后，就可以在 Object（物体）面板中找到刚才新建的按钮了。单击按钮，其中对应注册的操作符也将被调用，在视图中创建一个等边四面体，效果如图 11-28 所示。





图 11-28 运行注册在按钮中的 Operator（操作符）模块

由于创建的操作符已经注册进当前工程的正文中，所以也可以直接在视图窗口中，单击空格利用查询工具调用执行操作符，如图 11-29 所示。

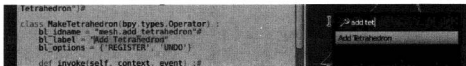


图 11-29 直接在视图中调用注册的 Operator（操作符）

## 11.3 内置脚本

在了解 Blender 脚本的基本 API 原理后，接下来就简单地介绍一下 Blender 中常用的几种内置脚本。

### 11.3.1 插件脚本

Add-Ons（插件）脚本是 Blender 中自定义扩展型脚本的集合组件，可以在 User Preferences（用户参数）面板中找到 Add-Ons 的配置界面，如图 11-30 所示。



图 11-30 Add-Ons（插件）脚本的配置界面

### 11.3.2 安装插件脚本

在配置界面中可以找到已集成的功能脚本，默认状态下它们都需要手动激活才能开启使用功能，激活时只需要单击对应脚本上的 Enable Add-On 选项即可。也可以自定义安装一些新的脚本，只需要将脚本文件 \*.py 存放在本地硬盘，单击 Install Add-On (安装) 选项，在弹出的文件管理器中选择脚本文件即可，如图 11-31 所示。当然也可以直接将脚本复制到 Blender 的目录 scripts/addons 下，使系统在启动时自动读取当前目录下的脚本。完成了脚本的安装和设置后，只需要单击 Enable 就可开启脚本功能了，使用 Save As Default (保存为默认) 选项，即可在下一次启动时自动激活当前的脚本设置。

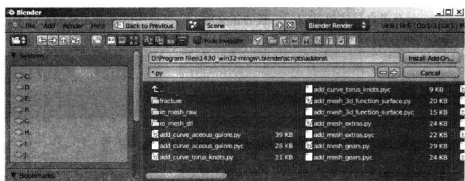


图 11-31 安装 Add-Ons (插件) 脚本

### 11.3.3 齿轮生成脚本

如图 11-32 所示，为 Add Mesh: Gears (齿轮生成) 脚本的属性描述，这里可以找到在视图中加载当前工具的路径，为 Add (新建) 菜单中的 Mesh (网格) 类目录。

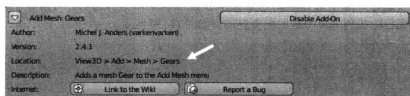


图 11-32 Gears (齿轮生成) 脚本的属性描述

转至 3D 视图中，单击 Shift + A 在 Mesh 菜单中即可找到新开启的 Gears 元件调用菜单，如图 11-33 所示。

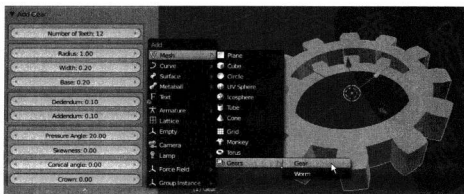


图 11-33 从新建菜单中调用齿轮元件

单击 Gear（齿轮）选项，即可在视图中添加一个齿轮物体。单击 T 在弹出的工具面板内，可找到相关的属性设置。

**Number of Teeth:** 齿数，用于定义轮齿的数量。

**Radius:** 半径，齿轮的半径值。

**Width:** 齿宽，也就是齿轮的厚度值。

**Base:** 轴面，这里的数值不能大于齿轮的半径。

**Dedendum:** 齿根的长度参数。

**Addendum:** 齿间的长度参数。

**Pressure Angle:** 压力角的度数。

**Skewness:** 齿面的斜度。

**Conical angle:** 齿轮的锥角。

**Crown:** 齿轮的冠型参数。

当完成了这些参数的设置之后，即可在场景中实时地预览新建的齿轮物体了。如果不使用 Add-Ons 功能，齿轮的建模可能需要花费大量的时间，而且还不一定能保证结果的准确。如图 11-34 所示，为使用脚本快速创建的齿轮组效果。

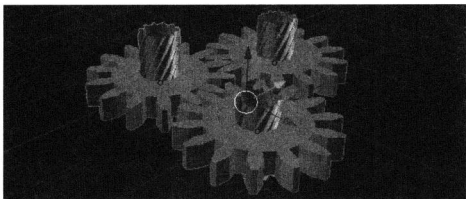


图 11-34 快速创建的齿轮组效果

### 11.3.4 云生成脚本

利用 Cloud Generator（云生成）脚本，可以快速地制作出漂亮的云朵效果。如图 11-35 所示，为其元件的属性面板。

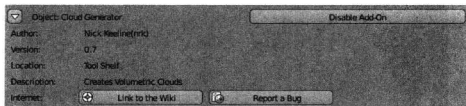


图 11-35 Cloud 云生成脚本的属性描述

首先在 3D 视图中随意添加几个 Cube 网格物体，如图 11-36 所示，使用快捷键 S 缩放它们的尺寸大小，并利用快捷键 G 将它们分别移动至任意的位置上。

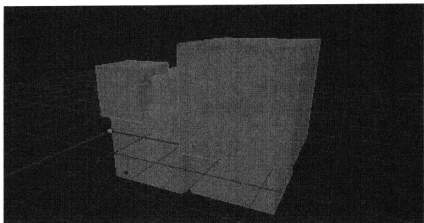


图 11-36 建立基础网格物体

单击 A 全选网格物体，然后使用快捷键 T 转至工具面板，可以看到这里新增了一项 Cloud Generator 菜单，如图 11-37 所示。

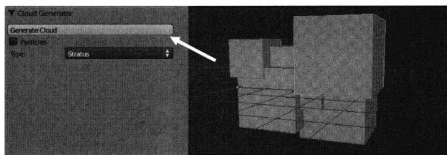


图 11-37 Cloud Generator（云生成）菜单

在 Type 下可以选择不同的云类型，激活 Particles（粒子）选项可基于网格产生粒子效果。这里保持默认配置，直接单击 Generate Cloud 按钮，云朵的效果如图 11-38 所示。

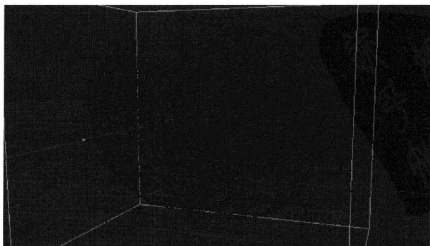


图 11-38 由脚本生成的云物体

网格将被脚本自动转换成可渲染的粒子物体，并添加了一个 Volume（体）材质，只需要再添加一个灯光，然后单击 F12 即可渲染出一个漂亮的云朵了，如图 11-39 所示。

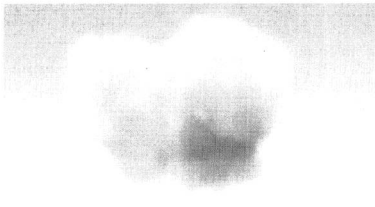


图 11-39 云的渲染效果

## 11.4 外置程序脚本

Blender 的 Python 脚本另外一个重要功能，就是可用于支持与外置程序<sup>①</sup>之间的转换与调用。其调用工作原理是使用 Blender 的 Render（渲染）等 API，将外置渲染器等程序与 Blender 的数据输出接口结合在一起，把场景直接导出至外置程序中执行编辑与运算。接下来，就简单地介绍几款常用的 Blender 外置渲染引擎。

### 11.4.1 YafaRay

YafaRay 是一款开源的光线追踪渲染引擎，同 Blender 一样，用户可以直接访问并使用它的源代码。YafaRay 的前身是由 YafRay（Yet Another Free RayTracer）发展而来，由于 Blender 社区的需求，在 2004 年 8 月 YafRay 被作为一款内置脚本集成到了 Blender 2.34 中。

作者 Mathias Wein 在 2005 年时，决定从零开始重写 YafRay 的源码，新的版本于 2008 年 10 月 1 日正式发布，并更名为 YafaRay。可以登录它的官方网站<sup>②</sup>了解更多的相关资料。



图 11-40 Kellyq—Honda NAs

YafaRay 利用了 Blender 的 ID Property Mechanism 序列属性机制，没有增加任何外部代码到 Blender 中，即可实现检索工程中的场景数据，并直接通过脚本配置将场景导入至 YafaRay 中执行渲染计算。

首先来看看使用 YafaRay 渲染的优秀作品，如图 11-40 所示。

在安装 YafaRay 之前需要先安装 Python 2.52 和 MSVC ++2005SP1 升级包，接着在 YafaRay 的官方网站中下载对应操作系统的安装文件，以及 YafaRay 的导出脚本文件。如图 11-41 所示，为 YafaRay 在 Blender 中的渲染面板。

① 例如，RenderMan、MentalRay 和 Thea Render 等外置渲染器。

② 网站地址：<http://www.yafaray.com/home>。

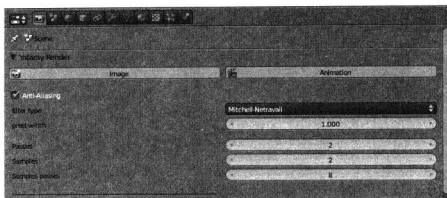


图 11-41 YafaRay 渲染面板

有关 YafaRay 的具体使用，将在第 15 章中重点讨论。

### 11.4.2 LuxRender

LuxRender 是一款物理真实的开源渲染软件，作为一款免费软件，任何个人和商业都可自由地下载使用。早在 2007 年，由 Terrence Vergauwen 带领的一个小型团队决定制作一款稳定的渲染器，并将它命名为 LuxRender。在 2008 年 6 月，它们正式推出了 LuxRender0.5，这也是第一次稳定版本的发布。当前版本的输出脚本仅支持 Blender 和 Maya，其他软件的输出脚本还正在开发中。

LuxRender 基于学院性质的 PBRT 项目，采用了由 Matt Pharr 和 Greg Humphreys 编写的非偏移光线追踪模拟算法。但是特别之处在于，它更是一款专注于产品渲染和艺术效果的渲染器，而非学术和科学目的的程序。因此它的开发团队将定期更新代码，用于改进渲染的速度，并添加更多的新特色功能。如果希望了解相关信息，请参考其官方网站<sup>①</sup>。来看看一幅由 LuxRender 渲染的效果图，如图 11-42 所示。



图 11-42 Walter Zatta—Veranda

① 网站地址：<http://www.luxrender.net>。

### 11.4.3 Indigo

Indigo 也是一款物理真实的渲染引擎，它建立在 MLT（Metropolis Light Transport）反向和双向路径追踪的基础上，而非利用传统的光线追踪，因此它能实现比传统光线追踪更真实的效果。为了追求一种无偏属性的计算结果，Indigo 会累加地执行渲染，以纯物理的方式来渲染场景，直到图像上的噪点逐渐被收敛至一个合理的范围，如果用户不手动终止的话，渲染将一直进行下去。考虑到它动辄几个小时的渲染工作效率，所以它更适合渲染静态画面而非动画，除非使用大型的渲染农场。它的工作原理也使得它不适用于渲染风格化图像，例如卡通着色等。

Indigo 不是免费软件，可以登录它的官方网站<sup>①</sup>完成订购。

接下来就看看 Indigo 带来的真实渲染输出效果吧，如图 11-43 所示。



图 11-43 Dcm—Living room closeup

### 11.4.4 V-Ray

V-Ray 是由 Chaos Group 开发的一款十分流行的渲染软件，也是在行业应用中最受欢迎的渲染引擎之一。它的渲染速度快、效果好，常被应用于室内外的建筑效果制作。2008 年，俄罗斯人 Andrey M. Izrantsev 开始主持 V-Ray 与 Blender 的兼容脚本开发，可以直接到它的官方网站<sup>②</sup>上下载最新版本脚本。Blender 的 V-Ray 脚本基于 V-Ray 的 Maya 独立版本，因此在使用时还需要购买最新版的 V-Ray for Maya。

来看看使用 V-Ray 完成的 Blender 作品，如图 11-44 所示。

① 网站地址：<http://www.indigorender.com>。

② 网站地址：<http://blender.bevice.ru/>。



图 11-44 Bertrand Benoit—Dinner

#### 11.4.5 SunFlow

SunFlow 是一款照片级真实图像合成的开源渲染引擎，它使用了一个灵活的光线追踪内核和一个可扩展的面向对象设计结构。由于它使用 Java 语言编写，为了能正常运行渲染器，还需要安装 JDK1.5 以上的 Java 配置软件包。可以从它的网站<sup>①</sup>上下载最新版本的渲染器和 Blender 下使用的导出脚本。如图 11-45 所示，为使用 SunFlow 渲染的优秀作品。



图 11-45 David Gudelius—Sponza

① 网站地址：<http://sunflow.sourceforge.net/>。



### 11.4.6 Octane

Octane 是世界上第一款基于 GPU 运算和物理非偏移算法的渲染器，它的核心开发成员 Brecht van Lommel 曾多年担任 Blender 的全职研发。前面介绍的 LuxRender 和 Indigo 也是基于物理非偏移的算法，它们都能提供十分真实的渲染结果，但是付出的代价是相当漫长的渲染时间。Octane 最大的特点就是它不再使用 CPU 来计算渲染，而采用 GPU 为非偏移运算带来更精确的结果，并且只需要花费很短的时间，经过测试最快可提升 5000% 的渲染效率！同时 Octane 还支持实时的渲染计算，允许用户在高质量的场景中与模型做即时的互动操作。可以登录 Octane 的官方网站<sup>②</sup>去订购最新版本。下面来看看 Octane 带来的惊人效果吧，如图 11-46、图 11-47 所示。

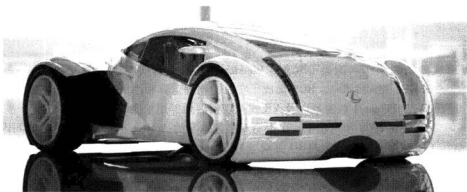


图 11-46 Artur Szymczak—Lexus

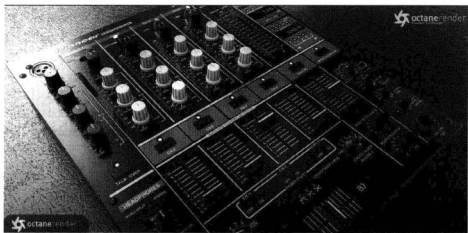


图 11-47 Eric—Pioneer

除了这些典型的外置渲染应用，Blender 的脚本集成还涵盖了各种领域下的功能调用。例如它可与 After Effect 集成实现后期视频的特效导出制作，或者配合 Shake 来完成镜头跟随的制作，如果与 Unity 3D 集成还能开展游戏开发和 iPhone 应用的制作，与 CSS 集成可用于实现网页上的脚本动画等。

② 网站地址：<http://www.refractivsoftware.com/>。

## 第 12 章

# 游戏引擎

除了强大的动画制作和渲染等功能，Blender 还拥有一套实时的动画开发工具——Blender Game Engine (BGE, 游戏引擎)。BGE 集成了实时的物理运算和逻辑运算引擎，可以开发电脑游戏、PS 电视游戏、iPhone 手机游戏以及 VR 虚拟现实展示等用于实时表现的 3D 视觉产品。

Blender 基金会在 2008 年组建了一个拥有 6 名艺术家的 Apricot 小队，在 6 个月时间内使用这套 BGE 制作了全球第一款全开源的 3D 游戏《Yo Frankie!》，如图 12-1 所示。游戏使用了 Cystal Space 和 BGE 两种引擎，支持 Windows、Mac OS 和 Linux 平台，用户可以登录它的官方网站<sup>①</sup>下载游戏以及所有的源代码。

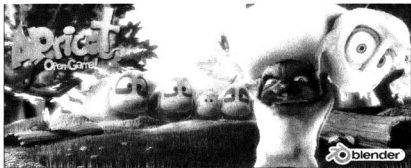


图 12-1 《Yo Frankie!》

## 12.1 BGE 开发环境

BGE 拥有十分强大的兼容性和可移植性，经过了多次的版本更新，BGE 目前可提供以下几项主要功能：

- ❑ 多平台兼容性，包括 GNU/Linux、Windows 和 Mac OS 等。
- ❑ Logic Brick 逻辑系统。
- ❑ 支持 Python 脚本系统。
- ❑ 支持 GLSL2.0 着色器规范。
- ❑ 支持 Bullet 等物理系统。
- ❑ 支持 WAV 等音频文件。

① 网站地址：<http://www.yofrankie.org/>。

- ☐ 支持实时阴影计算。
- ☐ 支持网络通信和关卡设计。

### 12.1.1 BGE 系统

前面曾简单地介绍过 Blender 集成的物理系统 Bullet，这是一款用于模拟自然物理效果的虚拟动力学库。要启动 BGE，需要将系统的运行环境切换至 Blender Game（游戏）模式，如图 12-2 所示。接下来将视图切换至 Game Logic（逻辑设计）布局，开始编写游戏脚本和逻辑设计。



图 12-2 切换至 BGE 游戏开发模式

#### 12.1.1.1 Game 游戏环境设置

Blender 的大部分面板都将被转换至游戏开发模式。如图 12-3 所示，为 BGE 下的渲染面板。

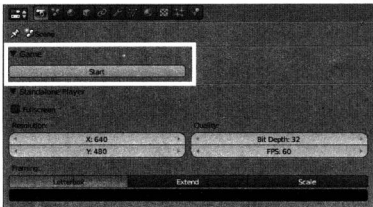


图 12-3 BGE 的渲染面板

渲染面板中取消 Render（渲染）选项，而更新为 Start（开始）游戏，单击后系统当前的 3D 视图将转换为游戏视图，用于运行已设计好的逻辑程序，也可以使用快捷键 P 快速地启动游戏的运行程序。

**Fullscreen：**是否开启全屏运行游戏模式，如果计划使用全屏效果，最好在游戏添加一个 Exit Game 退出游戏功能模块，其默认快捷键为 End，这样可避免一些系统在退出游戏后，依然保持游戏运行时的分辨率设置。

**Resolution：**设置默认全屏运行时的游戏画面分辨率。

**Quality：**设置游戏的运行性能，其中 Bit Depth（色彩深度）值用于控制游戏中的调色板数值，当设置为 8bit 时游戏将最多采用 256 色，而 FPS（帧率）则用于设置游戏运行时的最大刷新帧率。

**Framing：**用于设置游戏发布时画面中超出分辨率设置部分的显示模式，其中 Letterbox 模式将会把超出的部分填充为拾色器中选择的颜色，Extend 模式将适度地拉伸画面，而 Scale 模式将根据屏幕尺寸对画面做缩放输出显示。

#### 12.1.1.2 Stereo（立体）渲染选项

Blender 还支持 3D 的输出渲染。如图 12-4 所示，为 Stereo（立体）选项的参数面板。

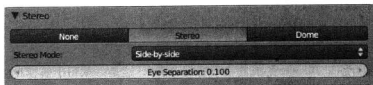


图 12-4 Stereo (立体) 选项的参数面板

以 Side by Side 模式为例,在这种模式下画面将根据 Eye Separation (眼距设置) 参数来调整视角差,当然这里还需要配合结点来执行合成,完成输出后就可以佩戴红蓝 3D 眼镜来观看立体特效了。如图 12-5 所示,为 3D 版《Elephants Dream》的渲染效果。



图 12-5 3D 版《Elephants Dream》

### 12.1.2 GLSL 高阶着色器

Blender 支持 GLSL (OpenGL Shading Language, 高阶着色器),它是由 OpenGL ARB 所建立,为 GPU 计算提供的一种直观可视算法。在渲染面板中,可以找到 GLSL 着色器的相关设置,如图 12-6 所示。

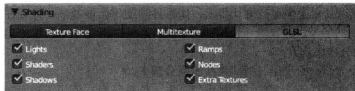


图 12-6 GLSL 着色器的设置

由于 GLSL 需要显卡和相关的驱动支持,所以如果用户使用的是较老版本的显卡,可能会在一些性能上或功能上缺乏硬件的支持,从而产生着色错误。可以在这里自定义地开启部分 GLSL 着色选项,以减少由于兼容性而导致的错误。以下是目前支持 GLSL 的主流显卡型号:

- ☐ ATI Radeon 9x00, Xx00, X1x00, HD2x00 以及 HD3x00 系列或最新的版本。
- ☐ Nvidia Geforce FX, 6x00, 7x00, 8x00, 9x00 以及 GTX 2x0 系列或最新的版本。

如果要开启材质的 GLSL 着色预览,需要在 3D 视图中单击快捷键 N 调出物体 Display (显

示) 属性, 并将 Shading (着色) 选项选择为 GLSL 模式, 如图 12-7 所示。

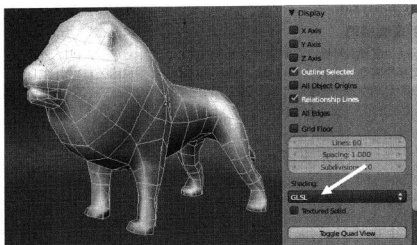


图 12-7 开启 GLSL 着色显示模式

这时只需要单击 Alt + Z 进入着色器预览模式, 然后调节灯光的强度和角度, 并旋转画面即可在场景中实时预览贴图 and 着色效果了。如图 12-8 所示, 物体上材质的反光面和高光面都会根据光线的变化而进行实时的调整, 这一切效果都是由显卡计算的 GLSL 直接完成的。

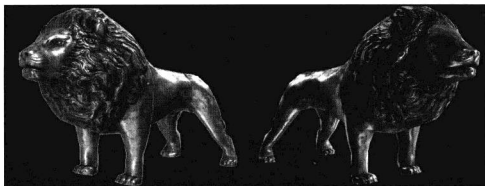


图 12-8 GLSL 预览效果

单击 P 进入游戏运行模式, 在 GE 引擎下的着色效果和预览中的着色效果会稍有差别, 如图 12-9 所示。这是因为 GLSL 只是一种由显卡运算的实时着色演算, 它仅用于表现材质表面在光照下的可视效果, 而在 GE 引擎演算时, 系统还会加入 RayTracing (光线追踪) 等高级着色技术, 而这些功能是显卡运算所无法提供的。



图 12-9 GLSL 预览和 GE 模式下的显示区别

但是这样的误差对画面的影响是很细微的，所以完全可以在 GISL 下直接实施游戏开发，并预览其在 GE 下的运行效果。

### 12.1.3 逻辑编辑器

Logic Editor（逻辑编辑器）是游戏开发工具的核心，它主要用于编写游戏的后台逻辑，以及人机之间的交互控制等。如图 12-10 所示，为 Logic Editor（逻辑编辑器）的界面。

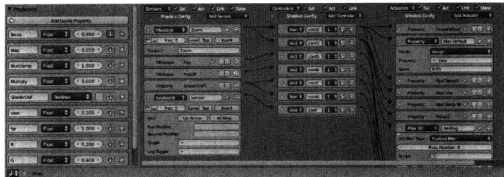


图 12-10 Logic Editor（逻辑编辑器）的界面

在 Blender 中使用 Logic Block（逻辑模块）来表示一个事件功能，一个事件逻辑模块又可以被分成三部分。其中，Sensors（触发器），用于识别事件的发生，例如球体落地的事件，或者敲击键盘的动作事件；Controllers（控制器），用于接收或监听事件的发生，并发出请求给系统，等待返回的逻辑响应；Actuators（促动器），它将接收 Controllers（控制器）发送的请求，并返回对应的响应值。它们三者之间以结点的形式连接在一起，实现数据的管线传递结构。这里要注意的是，不能将 Sensors 跨越 Controllers 直接连接到 Actuators 上，如果需要让一个事件在一个动作后被触发，可以考虑使用 Actuators 属性的 Sensors（触发器）。

#### 12.1.3.1 Property（属性）菜单

Logic Editor 不仅能用于管理逻辑模块之间的连接，单击 N 还可调出变量的 Properties（属性）菜单，如图 12-11 所示。单击 Add Game Property 按钮可添加一个新的变量，如果激活变量值旁边的叹号图标，可开启系统的 Debug（调试）打印。



图 12-11 Properties（属性）菜单中的变量管理

变量类型定义了逻辑模块中使用的变量属性，常用的变量类型如下。

**Timer:** 计数变量，用于计数或计时操作，例如用于记录每一个关卡中玩家所花费的时间。

**String:** 字符串型变量，用于存储文字等字符串信息，最多支持 128 个字符。

**Float:** 浮点型变量，用于存储 -10 000.000 到 10 000.000 范围内的浮点型数据。

**Integer:** 整型变量，用于存储 -10 000 到 10 000 范围内的整型数据。

**Boolean:** 布尔变量，用于存储 True 和 False 逻辑，对事件做判断操作，例如灯光的开关事件。

这些变量可被逻辑模块调用，也可被 Python 代码调用。当变量存储于逻辑模块中时，可用于参与逻辑和事件判断，而用于代码中时，则可存储人物的生命值和姓名等信息。

### 12.1.3.2 Node 逻辑模块

在 Logic Editor（逻辑编辑器）中，逻辑的计算是从左至右顺序执行的。Sensors（触发器）只有数据输出，因此它在右端只有一个输出接口，Controllers（控制器）起请求和响应的交换控制功能，所以在左右两端分别拥有一个数据输入和数据输出接口，Actuators（促动器）仅接收事件的请求，并从原输入接口返回请求的处理结果，所以它只在结点的左端有一个数据输入接口。可以使用鼠标的 LMB，按照 Node（结点）系统中连接结点的操作，将两个逻辑模块连接起来即可，对于一些多重响应的事件逻辑，模块间还支持一对多的连接方式。

Sensors 的触发事件是基于游戏中物体状态的，因此每一个物体都需要为其添加一个逻辑模块，用于探测它在游戏运行时的状态。当选择一个物体时，Logic Editor 将自动切换至当前物体的逻辑模块编辑模式，并且默认以物体的名称来命名模块名。当然还可以同时选择多个物体，将它们之间的逻辑做交叉连接，或者建立并列的逻辑调用，以产生多个物体在不同触发状态下联动响应的效果，如图 12-12 所示。

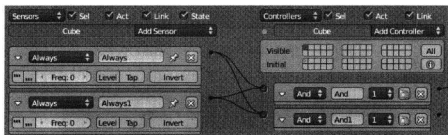


图 12-12 多模块连接

## 12.1.4 Hello Game!

### 12.1.4.1 建立场景

首先建立一个简单的游戏场景，来感受一下 BGE 开发的流程。单击 Shift + A 在场景中添加一个 Cube 物体和一个 Plane 物体，调整它们的位置，如图 12-13 所示。

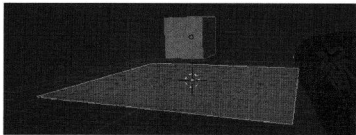


图 12-13 添加场景

将 Cube 物体移动至 Plane 的正上方，如果需要编辑物体的外形，可单击 Tab 进入其 Edit（编辑）模式，而不要尝试在 Object（物体）模式下直接执行缩放等形变操作。因为 BGE 下的物理计算是根据物体的边界属性来定义的，所以在 Object（物体）模式下执行的缩放可能会产生运算错误。

接下来需要修改物体的属性，BGE 中的所有新建物体都会被默认设定为 Static（静态）类型，也就是说，它们在 BGE 启动后将在原位置上保持静止不动。接着选择 Cube 物体，在物理面

板中，将其类型修改为 Rigid Body（刚体），如图 12-14 所示。

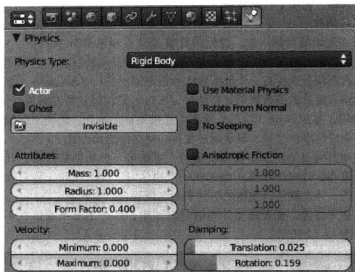


图 12-14 修改 Cube 为 Rigid Body（刚体）物体

同时设置物体的 Collision Bounds（碰撞边界）为 Box（矩形），如图 12-15 所示。



图 12-15 设置 Collision Bounds（碰撞边界）的形状

接下来为场景添加一盏 Sun 灯光，然后单击 P 运行游戏。这时就可以看到 Cube 将做一个自由落体运动，在接触 Plane 物体后停留在其表面上，效果如图 12-16 所示。

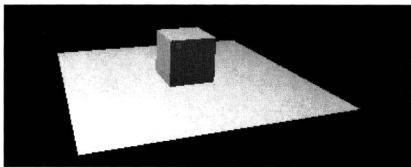


图 12-16 Cube 停留在 Plane 物体上

#### 12.1.4.2 创建事件逻辑

单击 ESC 退出游戏模式，再将视图切换至 Game Logic 中，在选择 Cube 的情况下，为其添加一个 Sensors（触发器），并将触发条件设置为 Keyboard（键盘）。接着在 Key 按钮中单击一次，系统会要求再次输入一个键盘按键，这时按方向键 Up Arrow。最后再为触发器添加一个 Controllers（控制器），设置逻辑关系为 AND，然后使用鼠标 LMB 将两者连接起来，如图 12-17 所示。



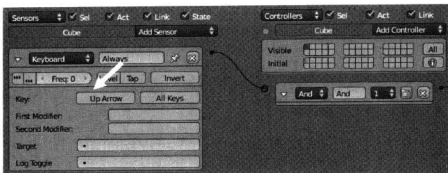


图 12-17 添加 Sensors 触发器

单击 Add 按钮为 Cube 添加一个促动器，将事件的响应类型设置为 Motion（运动），将 Loc 中的 Y 坐标参数修改为 0.1，并激活旁边的 L，使动作仅作用于本地坐标，否则物体将始终沿着全局 Y 的坐标方向前进。最后使用 LMB 将其与 Controllers 连接起来，如图 12-18 所示。

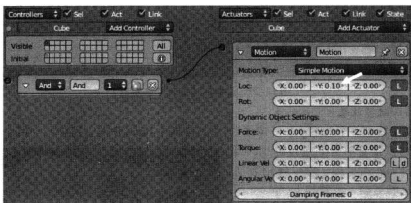


图 12-18 连接 Actuators 的事件响应

这时再次单击 P 进入游戏模式，当物体掉落在 Plane 表面后，每单击一次键盘的 Up 键，物体都会自动向其本地 Y 轴方向前进 0.1 个 Blender 单位，如果长按键盘，Cube 将会持续前进，直到滑出 Plane 继续下跌，效果如图 12-19 所示。

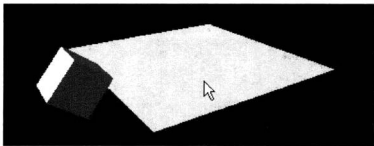


图 12-19 使用键盘控制 Cube 的移动

接下来为这个物体再添加两个 Sensors（触发器）。分别将其事件响应设置为：使用键盘 Left（左）方向键来控制物体的左转动作，使用键盘 Right（右）方向键来控制物体的右转动作，如

图 12-20 所示。单击 P 运行游戏后, 就可以使用左右方向键来控制物体的转向运动了。

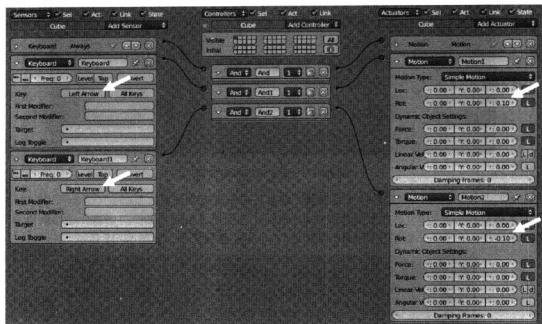


图 12-20 设置物体的转动触发器

#### 12.1.4.3 添加碰撞效果

再在场景中添加一个 IcoSphere (球体), 将其转换为 Rigid Body (刚体), 并设置碰撞属性为 Sphere (球体)。接着单击 P 运行游戏, 当 Cube 坠落到 Plane 面上后, 可以使用方向键盘来控制它的运动, 试着和 IcoSphere 产生碰撞互动, 并将其从平台上面推下去, 最后的效果如图 12-21 所示。

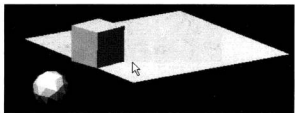


图 12-21 物体间的碰撞互动

#### 12.1.4.4 调用 F-Curve (曲线) 动画

当然也可以结合 F-Curve (曲线) 来自定义物体的响应运动, 首先为 Icosphere 球体添加一个简单的动作。在第 0 帧的时候添加一个 Loc 关键帧, 在第 20 帧的时候将球体向 Z 正轴移动 4 个 Blender 单位, 再添加一个 Loc 关键帧, 如图 12-22 所示。



图 12-22 为球体添加一个曲线动画

在 Logic Editor (逻辑) 面板中, 为 Icosphere 球体添加一个由键盘控制的事件, 当单击 Space (空格) 键时, 促动器将调用球体的运动 F-Curve 曲线动画, 具体设置如图 12-23 所示。

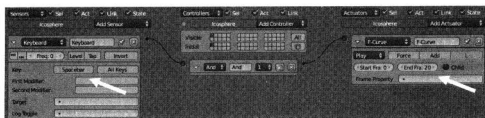


图 12-23 使用促动器来调用物体的自定义曲线动画

最后单击快捷键 P, 在游戏模式中通过单击空格键, 即可控制球体的自定义运动了。如图 12-24 所示, 每单击一次 Space 按钮, 球体都会按照曲线定义的动画, 向上跃起 4 个 Blender 单位的距离。

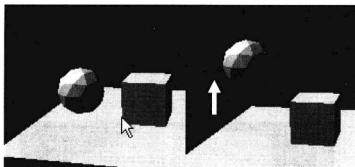


图 12-24 使用键盘事件来激活物体的曲线动画

## 12.2 物理属性

在 BGE 中, 需要为每一个对象都指定对应的物理属性。物体的物理属性可以在 Physics (物理) 面板中定义, 如图 12-25 所示, 为物体 Physics (物理) 属性的参数面板。

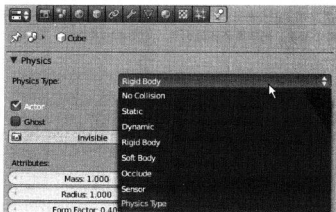


图 12-25 物体 Physics (物理) 属性的参数面板

在选择物体的物理属性后，还需要为其指派场景中的角色类型。

**Actor:** 角色，如果当前物体被设定为 Actor（角色），它将与周围的环境产生交互，并产生触发事件或者执行响应。对于刚体和软体等需要与场景互动的物体，必须将其设置为 Actor（角色）类。

**Ghost:** 鬼魂，这种类型的物体不会与物体产生交互，Actor 可以直接从其内部穿过。

**Invisible:** 隐形，处于 Invisible（隐形）状态下的物体，将在游戏中处于不可见状态。

### 12.2.1 无碰撞类

碰撞是物体间执行互动运算的一种效果，如果设置当前的物体为 No Collision（无碰撞）类型，那么系统将取消对其碰撞的检测和计算。当任何物体与它接触时，它们之间将不会产生任何阻挡或反弹等效果。

碰撞将消耗较多的 CPU 计算，所以使用 No Collision（无碰撞）属性可以在一定程度上提升系统场景的运算性能。如图 12-26 所示，为物体选择 No Collision（无碰撞）类型的设置面板。

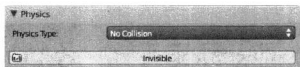


图 12-26 设置物体属性为 No Collision（无碰撞）类型

还是使用刚才的场景，只是这次将 Plane 的属性设置为 No Collision。单击 P 运行游戏后，可以发现 Cube 将直接穿过 Plane 继续下落，效果如图 12-27 所示。

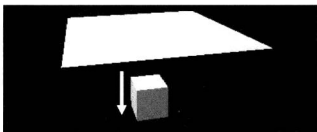


图 12-27 无碰撞属性的物体将产生穿越效果

### 12.2.2 静态类

Static（静态）类型用于制作静止物体，使其不会受其他物体的碰撞等作用效果而改变状态。常用于组建游戏的整个场景，例如墙体和地面等。如图 12-28 所示，为 Static（静态）物体类型的属性面板。

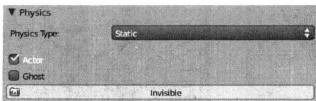


图 12-28 Static（静态）物体属性

### 12.2.3 动态类

一个 Dynamic（动态）类型的物体将根据物理环境的设置去参与模拟运算，例如接受重力的影响或其他物体的碰撞。可以使用快捷键 Ctrl + Alt + Shift + P 将其运动路径记录为曲线数据，便

于回放使用。如图 12-29 所示，为 Dynamic（动态）类型的参数面板。

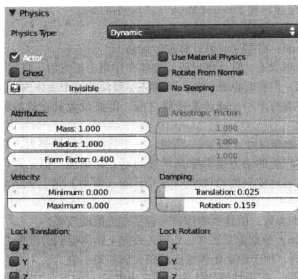


图 12-29 Dynamic（动态）类型的参数面板

#### 12.2.3.1 Physics（材质物理）属性

单击 Use Material Physics 可开启材质的物理属性，当为一个物体添加材质后，可在其材质面板中找到对应的物理属性面板，如图 12-30 所示。



图 12-30 Material Physics（材质的物理）选项

这里的材质物理动态效果，可根据材质的属性来定义物体间的动态交互作用效果。

#### 12.2.3.2 Radius（边界范围）

Radius（边界范围）用于定义一个物体的碰撞边界范围。在物理引擎的计算中，CPU 不会去读取一个物体的网格模型，而是使用一个基础物体形状的边界范围来判断碰撞的边缘。如图 12-31 所示，为不同 Radius 半径值下物体的碰撞边界。

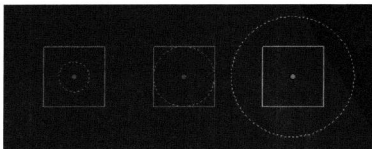


图 12-31 Radius（边界范围）

Radius 只控制边界的范围，而形状则由 Collision Bounds（碰撞边界）类型来控制，默认的边

界形状为 Box（矩形）。可供选择的类型还包括 Sphere（球体）、Cylinder（圆筒形）、Cone（圆锥形）、Convex Hull（凸壳形）和 Triangle Mesh（三角网格形）。

### 12.2.3.3 其他参数

**Rotate From Normal:** 根据法线来计算翻转运动。

**No Sleeping:** 无休眠状态，默认情况下，物体在作用力消失后，将被物理引擎从运算序列中移出，以节约系统资源。但有时也有例外，可能还需要保持这个物体的运算进程，例如足球场上的足球，当它落地后并等待下一个球员来接球之前，需要设置其在这段时间之内为 No Sleeping（无休眠）状态。

**Attribute:** 物体的物理属性，如 Mass 质量，一个物体的质量越大，移动它就需要更大的外力，同时它对其他物体产生的动力也会根据质量的增大而加大。但是再大的质量也不会使物体下落得更快，要改变下落速度只能修改全局的 Gravity（重力）值，或者修改 Damping 中的空气阻力参数。

**Form Factor:** 形态参数，值越高物体越不容易产生滚动效果。

**Anisotropic Friction:** 分别定义了 X、Y 和 Z 方向上物体受到的摩擦力。

**Velocity:** 速度极值，用于约束物体速度的最大值和最小值。

**Damping:** 阻力参数，用于控制物体的运动自由度。

**Lock Translation/Rotation:** 用于锁定物体在 X、Y 和 Z 轴上的移动和转动运动。

## 12.2.4 刚体类

Rigid Body（刚体）是在 Dynamic 物体的基础上，增加了边的刚性碰撞计算，这和在物理模拟中讨论过的 Rigid Body 原理一样。如图 12-32 所示，为 Rigid Body（刚体）物体的参数面板。

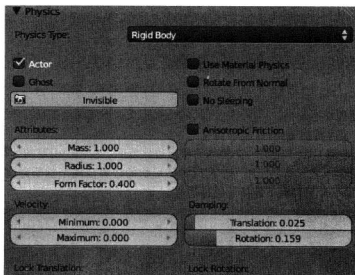


图 12-32 Rigid Body（刚体）物体的参数面板

## 12.2.5 软体类

Soft Body（软体）则是在 Dynamic 物体的基础上，添加了弹性边的运算逻辑。如图 12-33 所示，为 Soft Body（软体）的参数面板。

软体的控制参数和物理模拟中的设置类似，请参考第 9 章中对于相关参数的描述。

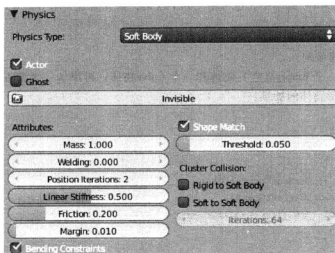


图 12-33 Soft Body (软体) 的参数面板

## 12.2.6 闭塞类

### 12.2.6.1 闭塞原理

当设置一个物体类型为 Occlude (闭塞) 后, 任何在其背后的物体都将无法被渲染出来, 通常在制作墙体或者山体时, 可以将墙体设置为 Occlude。它最重要的功能就是可以减少场景的运算量, 不用将在其后面不可见部分的多边形发送给 GPU 运算。例如将一个房屋设置为 Occlude 物体, 当镜头在房屋外时, 里面的家具等物体将因为被房屋遮挡了视线, 而不会被渲染出来。当进入房间时, 房屋的墙壁将不再遮挡镜头的视线, 屋里的家具摆设这时才会被渲染并呈现在眼前。

在使用 Occlude 时还应注意, 如果 Occlude 类型的物体尺寸过小, 它不仅无法遮挡任何其他物体, 反而会成为 CPU 的负担。所以应合理地设置可预见的遮挡场景, 如果当前的场景不适合 Occlude, 应考虑舍弃这种运算模式。例如在大场景中的大型建筑和山脉, 可以使用它, 但是如果是较多重叠物体的场景或是高模面数的物体, 应尽量减少使用它。

在默认情况下, Blender 的 Occlusion 运算是开启的, 如果场景中的物体超过 100 个, 开启 Occlusion 可提升至少 5 倍以上的运算速度, 如图 12-34 所示。也可以在 World (世界) 面板的 Physics (物理) 环境设置中, 自定义关闭 Occlusion 计算。

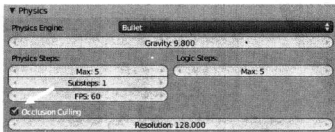


图 12-34 开启 Occlusion (闭塞) 选项

Resolution (分辨率) 用于设置闭塞的精度, 对于具有较高硬件性能的系统, 可以设置 Resolution 为 1024 以上。

在逻辑模块或 Python 脚本中，也可使用如下代码，来开启或关闭 Occlusion 运算。

```
obj.occlusion = True
obj.setOcclusion(True,False)
```

#### 12.2.6.2 闭塞应用实例

在场景中新建一个 Empty（空）物体和一个 Cube 物体，再添加一个脚本 Create.py，用于创建一个充满随机矩形物体的场景。

```
import random
import GameLogic
import bge

cont = GameLogic.getCurrentController()
scene = GameLogic.getCurrentScene()
owner = cont.owner
cube = "Cube"

for i in range(0,10500):#设置添加物体的循环
    x = random.uniform(-41.3,41.3)
    y = random.uniform(-41.3,41.3)
    z = random.uniform(-41.3,41.3)
    owner.position=[x,y,z]
scene.addObject(cube,owner)#向场景中添加新建物体
```

接着在 Logic Editor（逻辑编辑器）中，添加 Empty（空）物体对脚本的调用控制，以及脚本对 Add Object（新建物体）促动器的调用控制，最后的设置如图 12-35 所示。



图 12-35 添加 Empty 对脚本的调用

在场景中添加一个镜头和一个 Plane 物体，单击 Tab 进入 Plane 物体的编辑模式，在其正中间挖一个洞，如图 12-36 左图所示。接着将 Plane 和镜头绑定为父子关系，这样在镜头运动时，Plane 将一直做跟随运动并处在镜头的前方，如图 12-36 右图所示。



图 12-36 添加镜头前的 Plane 物体

选择 Plane 物体，并设置其物体属性为 Occlude（闭塞）模式，如图 12-37 所示。

再选择摄像机，为其添加一个由 Space（空格键）控制的旋转运动，取消 L 本地运动约束，



如图 12-38 所示。



图 12-37 设置 Occlude（闭塞）属性



图 12-38 设置镜头的控制

单击 P 后就可以在游戏模式中查看 Occlude 效果了，如图 12-39 所示。可以看到脚本在场景中创建了大量的随机矩形物体，但是由于 Plane 开启了 Occlude 运算，在 Plane 后面的物体并没有显示出来，而只有在其边缘外或者孔中的物体才可以被正常渲染出来。可以单击 Space（空格键），通过旋转镜头来查看物体的遮挡显现效果。

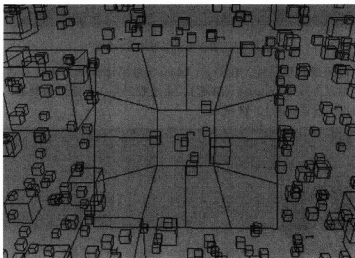


图 12-39 物体的 Occlude 运算效果

## 12.2.7 触发类

### 12.2.7.1 触发类原理

Sensors（触发）类物体就像雷达一样，可用于感知静态或动态物体是否触发了一个事件，并根据触发事件来产生判断逻辑。通常可以应用它来测试物体之间是否产生了碰撞，例如使用 Sensors 来检测角色的脚是否接触到地面。如图 12-40 所示，为 Sensors（触发）类物体的参数面板。



图 12-40 Sensors (触发) 类物体的参数面板

### 12.2.7.2 触发类应用实例

建立一个如下的场景，在场景中添加两个 Static 类型的矩形，并在它们的中上方放置一个 Icosphere 球体，它的直径应大于矩形间的缝隙宽度。将球体设置为 Dynamic 类型，其 Collision 边界属性设置为 Sphere。在球体的下方再添加一个类平面的物体，将其属性设置为 Sensors (触发) 类，其 Collision 边界属性设置为 Triangle Mesh，并设置球体为其父物体，这样球体的运动将连动类平面的运动，效果如图 12-41 所示。

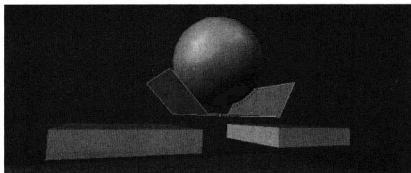


图 12-41 设置场景与物体属性配置

接着设置类平面的逻辑块属性，在 Logic Editor (逻辑编辑器) 中，为其添加一个 Sensors (触发器)，设置其触发事件为 Collision (碰撞)，由于它只是一个触发检测器，所以没必要添加 Controller 等逻辑结点，设置如图 12-42 所示。



图 12-42 为类平面添加一个 Sensors (触发器)

接下来为 Icosphere 球体设置逻辑块，首先添加一个由 Keyboard (键盘) 控制的 Sensors (触发器)，然后将它与前面 Plane 类平面的 Sensors 一起接入第二级的 Controller (控制器) 结点上，并促动一个向 Z 轴正向跃起的动作促动器，如图 12-43 所示。由于 Controller 是 AND 与逻辑属性，所以只有当类平面检测到存在碰撞事件，并同时有用户单击 Space (空格键) 的动作时，才能激发物体的跃起运动。例如如果制作的是一个人物的腿，那么使用这样的 Sensors 探测逻辑，就可以正确地控制角色只有在脚落地后才能允许做出起跳动作。否则如果只有一个键盘控制的起跳控制，而没有触发检测，就有可能出现腿还在半空中时就被强制执行起跳动作，这是明显不符合逻辑的设计。

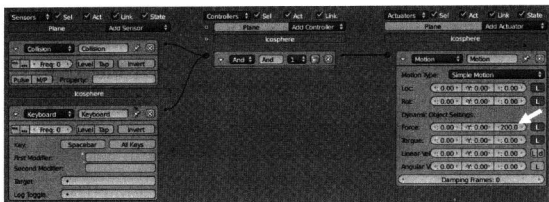


图 12-43 设置 Icosphere 球体的 Sensors 探测逻辑

最后单击 P 进入游戏模式，可以看到，只有当物体在下落并碰撞在矩形物体上时，才能被允许执行跃起动作，否则 Space（空格）操作将被视为无效，效果如图 12-44 所示。

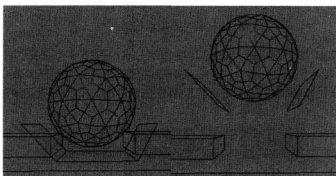


图 12-44 触发类的检测效果

## 12.3 触发器

Sensors（触发器）是一个事件的触发探测器，例如附近有靠近的物体，或者单击键盘的按键操作等。同时 Sensors 也是一个信号源，用于向 Controllers（控制器）发出状态指令。如图 12-45 所示，为一个 Sensors 节点的面板结构。

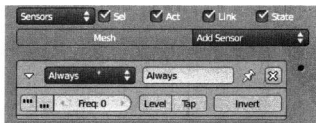


图 12-45 Sensors 节点的面板结构

### 12.3.1 触发器结构

一个 Sensors 的顶部有 4 个选项，激活这些选项可分别开启不同的选择模式。单击 Sel 可同时

显示所有被选择物体的 Sensors 结点, Act 则仅显示当前选择的物体 Sensors 结点, 激活 Link 选项将显示所有非空链接的 Sensors (触发器), 而 State 仅显示连接在当前活动 Controllers (控制器) 上的 Sensors。

单击 Add Sensors 按钮可添加一个新的 Sensors (触发器), 默认将以当前被选择激活的物体来命名。如图 12-46 所示, 单击以物体名称命名的按钮可收起该物体下的所有 Sensors (触发器) 结点。

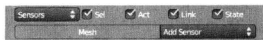




图 12-46 收起 Sensors 列表

Sensors (触发器) 有一个标题栏和一个属性栏, 如图 12-47 所示。标题栏可用于删除 Sensors (触发器)、选取触发类型或者定义 Sensors 名称。触发器结点的名称可被 Python 脚本调用, 因此需要保证命名的唯一性, 而最左方的三角按钮则是用于收起 Sensors 的属性显示。



图 12-47 Sensors 标题栏

Sensors 属性栏有多个属性按钮, 如图 12-48 所示, 这些按钮可分别表示当前结点的逻辑和数字, 其中激活  代表返回值为 True (真), 而激活  则代表返回值为 False (假)。当设置为 True 返回值时, 结点在被触发时将向控制器返回正值操作符, 如果弹起这个按钮, 结点在触发时将以负值输出。

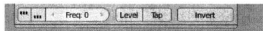


图 12-48 Sensors 属性栏

**Frequency (Freq):** 频率值, 用于设置从事件触发到逻辑输出的延时, 默认为 0, 表示没有输出延时。延时的时间可以根据频率值来计算, 例如默认逻辑模块的脉动频率为 60 Hz, 也就是每秒钟将产生 60 次探测触发, 所以如果设置为 130, 那么输出的延时时间为半秒钟。

**Level:** 是否使触发器工作在状态属性的事件上。

**Tap:** 和 Level 一样, 都用于逻辑的状态变化探测, 但两者不允许被同时激活。

**Invert:** 使 Sensors 反向输出。

Sensors (触发器) 还有多种状态可供选择使用, 如图 12-49 所示, 为 Sensors 的部分触发类型列表。

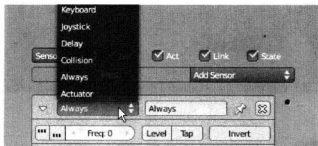


图 12-49 选择触发类型

## 12.3.2 常用触发器

接下来就简单介绍几种常用类型的触发器。

### 12.3.2.1 Delay (延时) 触发

Delay (延时) 触发可设置发送信号的延时时间, 如图 12-50 所示, 为 Delay (延时) 触发的逻辑模块。

**Delay:** 定义了发送 True 响应前的等待时间, 可按照频率值来计算。

**Duration:** 发送 False 响应前的等待时间。

### 12.3.2.2 Keyboard (键盘) 触发

Keyboard (键盘) 触发可用于探测键盘的输入, 并输出 True 响应。如图 12-51 所示, 为 Keyboard (键盘) 触发的逻辑模块。

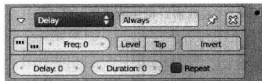


图 12-50 Delay (延时) 触发的逻辑模块

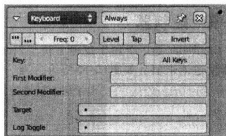


图 12-51 Keyboard (键盘) 触发的逻辑模块

**Key:** 可以在 Key 中输入一个控制触发的按键, 只有当这个按键被单击时触发器才会做出输出响应, 当选择 All Keys 时触发器将接受任意按键的触发事件。

**First/Second Modifier:** 组合键触发功能, 例如可以在这里添加一个组合键为 Ctrl + R 的触发事件。

**Target/Log Toggle:** 用于记录用户的输入值, 如果与 Property 触发器组合使用, 可以用来制作密码输入等功能模块。

### 12.3.2.3 Mouse (鼠标) 触发

Mouse (鼠标) 触发用于探测鼠标的动作, 如图 12-52 所示, 为 Mouse (鼠标) 触发的逻辑模块。

可以在下拉菜单中选择对应的鼠标动作, 例如 Mouse over (鼠标经过) 或 Middle button (鼠标滚轮) 动作等。

### 12.3.2.4 Property (属性) 触发

Property (属性) 触发, 用于探测物体的属性变化, 如图 12-53 所示, 为 Property (属性) 触发的逻辑模块。



图 12-52 Mouse (鼠标) 触发的逻辑模块

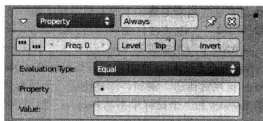


图 12-53 Property (属性) 触发的逻辑模块

模块的逻辑很简单,当 Prop 按照关系逻辑,例如当 Prop 的值 Equal (等于) Value 值时,触发器就会输出一个 True (正) 响应值。

### 12.3.2.5 Collision (碰撞) 触发

Collision (碰撞) 触发用于探测当前物体是否产生了碰撞事件,和它类似的触发类型还有 Touch (接触) 触发、Near (接近) 触发和 Radar (雷达) 触发等。如图 12-54 所示,为 Collision (碰撞) 触发的逻辑模块。

其他类型的触发器还包括 Always (长效) 触发,这个 Sensors 不与外界进行交互,将持续地产生一个 True 逻辑输出。

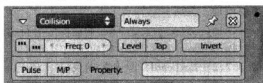


图 12-54 Collision (碰撞) 触发的逻辑模块

## 12.4 控制器

Controller 是一个用于逻辑中转与判断的计算控制器,它将根据触发器产生的响应向促动器发送请求值。如图 12-55 所示,为一个 Controller (控制器) 模块的属性面板。

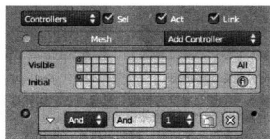


图 12-55 Controller (控制器) 模块的属性面板

Controller 有一个图层管理模式,单击属性面板上的 **1** 按钮,可将当前 Controller 移动至目标图层中。

### 12.4.1 控制器逻辑

Controller 主要执行逻辑的判断计算,如表 12-1 所示,为常用的逻辑判断计算结果。

表 12-1 常用判断逻辑的计算结果

Sensors 逻辑输入	AND 与	OR 或	XOR 异或	NAND 与非	NOR 或非	XNOR 异或非
A = B = C = 0	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
A = 1, B = C = 0	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE
A = B = 1, C = 0	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE
A = B = C = 1	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE

表 12-1 中左栏为 Controllers 的连接输入逻辑,右栏则为不同逻辑判断的计算结果。为了更好地理解这些判断关系,下面来简单地了解一下每种关系判断的原理。

#### 12.4.1.1 AND (与) 逻辑

AND (与) 逻辑是 Controllers 的默认关系逻辑,常用于最简单的触发与促动连接。当有多个 Sensors 连接至 AND 逻辑时,只有在所有 Sensors 同时产生 True 逻辑时,才能激活控制器的向下 True 正逻辑传递。例如,需要使用一个鼠标滑过并单击的动作,这时就需要添加一个鼠标滑过的 Sensors 和一个鼠标单击的 Sensors,然后同时连接至设置为 AND 关系 Controllers。可以用如下表达式来计算 AND 逻辑:

$$Output = A \cdot B \cdot C$$

### 12.4.1.2 OR（或）逻辑

当一个或多个 Sensors 连接至 OR 关系的 Controllers 时，只要其中至少有一个 Sensors 产生了 True 触发，Controllers 即可向下传递一个 True 正逻辑。例如可以设计一个使用快捷键 ESC 和 Q 两种方式来退出游戏的功能，这时只需要将两个 Keyboard 属性的 Sensors 连接至 OR 关系 Controllers 即可。可以用如下表达式来计算 OR 逻辑：

$$\text{Output} = A + B + C$$

### 12.4.1.3 XOR（异或）逻辑

XOR（异或）逻辑，仅当多链接中的一个 Sensors 传递 True 信号时，Controllers 才输出 True 逻辑，如果两个或多个 Sensors 同时输出相同逻辑，Controllers 则输出 False。可以用如下表达式来计算 XOR 逻辑：

$$\text{Output} = A \oplus B \oplus C = A \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C$$

### 12.4.1.4 NAND（与非）逻辑

NAND（与非）逻辑与 AND 的逻辑关系相反，只要有一个为 False 的 Sensors 输出，Controller 即输出 True 逻辑。可以用如下表达式来计算 NAND 逻辑：

$$\text{Output} = \overline{A \cdot B \cdot C}$$

### 12.4.1.5 NOR（或非）逻辑

NOR（或非）逻辑与 OR 的逻辑关系相反，只有当所有连接的 Sensors 逻辑都为 False 时，Controller 才输出一个 True 逻辑。可以用如下表达式来计算 NOR 逻辑：

$$\text{Output} = \overline{A + B + C}$$

### 12.4.1.6 XNOR（异或非）

XNOR（异或非）是 XOR 的一个反逻辑。可以用如下表达式来计算 XNOR 逻辑。

$$\text{Output} = A \oplus B \oplus C = \bar{A} \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot C + A \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot B \cdot C$$

## 12.4.2 Python 逻辑控制器

除了逻辑关系控制器，还可以使用 Python 来控制逻辑输出。接下来以一个简单的例子讲解 Python 逻辑控制器的控制方式。如图 12-56 所示，为 Python 逻辑类控制器的逻辑模块，也可以使用 Module 来调用脚本中的自定义模块。



图 12-56 使用 Python 脚本来控制逻辑关系

首先在场景中添加两个默认网格物体，一个 Icosphere 球体和一个 Cube（矩形），如图 12-57 所示。

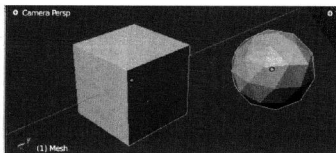


图 12-57 建立场景

接着进入 Logic Editor (逻辑编辑器), 在其中输入如下脚本, 命名并保存为 ModuleDemo.py。

```
import GameLogic
import Rasterizer

def showMouse():#定义鼠标显示控制
    Rasterizer.showMouse(1)

def run():#定义鼠标滑过动作
    cont = GameLogic.getCurrentController()

    if cont.sensors['mouse_over'].positive:
        cont.activate('spin')
    else:
        cont.deactivate('spin')
```

接下来同时选择两个物体, 在 Logic Editor (逻辑编辑器) 中分别添加触发器, 并设置 Mesh 的触发动作为 Mouse Over (鼠标经过), 如图 12-58 所示。



图 12-58 设置两个物体的触发属性

接着在 Controllers 中选择 Cube 物体, 并在其属性列表中添加两个控制器, 分别设置为 Python 控制器。将控制器 on Script 链接至 Cube, 并在模块调用中输入定义的方法 showMouse(), 这个方法可以直接在脚本中控制物体的状态。将 Mesh 的鼠标滑过触发器链接至 Cube 中的控制器 on Script 1 上, 并在模块调用中输入 Python 中的 run() 方法, 这个方法可以根据输入值来决定是否输出正向逻辑值, 最后的链接效果如图 12-59 所示。

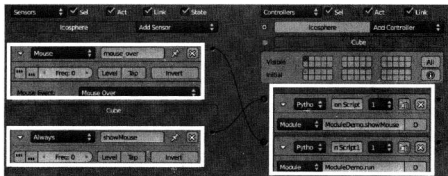


图 12-59 链接控制器



最后为 Cube 添加一个促动器，并将其命名为 spin，动作为围绕 Z 轴的旋转动作，将其与 on Script1 链接起来，如图 12-60 所示。这样通过在代码中对方法的逻辑定义，即可使控制器在探测到触发器 Mouse Over 为正逻辑输入时，激活 spin 促动器，运行旋转动作，反之将不会激活促动器来控制物体的旋转。

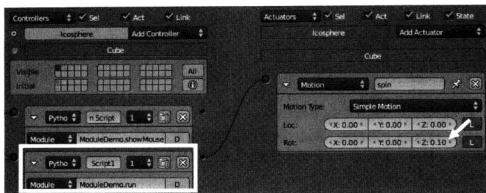


图 12-60 完成促动器的链接

单击 P 进入游戏模式，当把鼠标移动至 Mesh 表面上时，Cube 将在 Z 轴上旋转起来，直到鼠标从 Mesh 上移开，Cube 才停止转动运动，效果如图 12-61 所示。

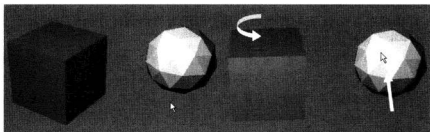


图 12-61 使用 Python 完成逻辑控制

## 12.5 促动器

Actuators（促动器）在游戏中用于输出动作，只有当控制器给出正逻辑指令时，促动器才会被执行。例如可用促动器来实现物体的移动和转动运动，播放声效和音乐事件等。Blender 提供了 18 种不同类型的促动器，接下来简单介绍一下其中常用的几种。

### 12.5.1 运动促动器

Motion（运动）促动器用于设定物体的移动和转动等运动，它有两种控制方式，一种是 Simple Motion，另一种是 Servo Control。如图 12-62 所示，为 Simple Motion 模式的促动器面板。

对于不同物理属性的物体，其 Motion 控制参数也会稍有不同，例如对于属性为 Static、No Collision 和 Occlude 类的物体，它们的促动器参数面板都相同。但是如果将属性修改为 Dynamic 或者 Rigid Body 类，促动器的参数面板将新增几个控制器，如图 12-63 所示。

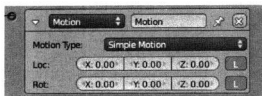


图 12-62 Simple Motion 模式的 Motion 促动器面板

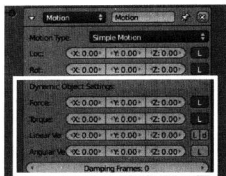


图 12-63 Dynamic 属性物体的 Motion 促动器面板

**Location (Loc):** 位移控制, 移动计量为 Blend 单位距离。

**Rotation (Rot):** 旋转控制, 每一转的计量为 7.2, 也就是说每转动一度的数值应设置为 0.02。

**Force:** 推动力, 能促动物体的运动。

**Torque:** 转动力, 使物体产生转动。

**Linear Velocity:** 线速度参数, add 按钮可将参数值叠加至物体当前的运动速度基数上。

**Angular Velocity:** 角速度参数, 用于提供旋转的角速度。

**Damping Frames:** 阻力值, 用于减缓物体的运动。

## 12.5.2 曲线动画促动器

F-Curve 促动器可在接收 True 逻辑触发后, 调用为当前物体存储的曲线动画。如图 12-64 所示, 为 F-Curve 促动器的参数面板。

该促动器支持 5 种 F-Curve 模式, 包括: Play (正序播放) 模式、Ping Pong (双向回放) 模式、Flipper (反向回放) 模式、Loop Stop (循环播放) 模式和 Property (属性播放) 模式。

**Force:** 强制播放曲线动画, 激活后面板中会出现一个 L 按钮, 用于将动画约束至本地坐标上运行。

**Add:** 由于物体的曲线数据被存储为全局坐标格式, 所以如果强制运行物体, 可能会跳回到原位置来执行动画, 当单击激活 Add 后, 物体将在当前的本地坐标位置上运行曲线动作。

**Start/End Frame:** 定义动画的回放初始帧位置和结束帧位置。

**Child:** 是否更新子物体的曲线动画。

**Frame Property:** 输入调用曲线的帧属性。

## 12.5.3 场景促动器

Scene (场景) 促动器, 用于控制场景, 以及是否连接或者跳跃至其他 Blend 工程下设计的关卡, 也常用于制作界面之间的转换等效果。如图 12-65 所示, 为 Scene (场景) 促动器的属性面板。

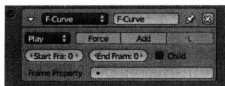


图 12-64 F-Curve 促动器的参数面板



图 12-65 Scene (场景) 促动器的属性面板

Scene (场景) 促动器在接受到 True (正逻辑) 信号后, 将对当前场景执行多种转换操作。

**Restart:** 重置当前场景。

**Set Scene:** 跳转至指定场景, 例如选择的某一个关卡场景。

**Set Camera:** 跳转至指定的镜头视图, 例如用于在赛车游戏中切换视角。

**Add Overlay Scene:** 在当前场景上叠加一个新场景, 例如单击快捷键可弹出人物背包中的物品列表。

**Add Background Scene:** 在场景下添加一个新背景, 例如为一个人物选择不同的背景图。

**Remove Scene:** 移除指定的场景。

**Suspend Scene:** 暂停当前的场景。

**Resume Scene:** 恢复暂停状态的场景。

## 12.5.4 编辑促动器

Edit (编辑) 促动器用于修改物体的网格结构, 以及添加或者删除物体。如图 12-66 所示, 为 Edit (编辑) 促动器的属性面板。

Edit (编辑) 促动器有多种物体编辑模式。

**Add Object:** 添加一个新物体, 例如制作枪管中射出的子弹。

**End Object:** 让当前物体从场景中消失, 例如制作士兵战死后, 身体从战场上消失。

**Replace Mesh:** 使用另一个物体来替换当前物体, 例如制作马里奥在吃了蘑菇后的变身效果。

**Track to:** 让当前物体跟随另一个物体运动, 例如让枪管始终对准画面中的十字准星位置。

**Dynamic:** 动态的改变物体属性, 例如修改物体的质量。

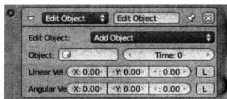


图 12-66 Edit (编辑) 促动器的属性面板

## 12.5.5 镜头促动器

Camera (镜头) 促动器用于制作镜头的跟随等动作, 如图 12-67 所示, 为 Camera (镜头) 促动器的属性面板。

**Camera Object:** 输入镜头将跟随的目标物体名称。

**Height:** 定义镜头离物体中心的水平高度。

**Min:** 定义镜头与物体的最小距离值。

**Axis:** 将镜头约束为跟随物体的 X 轴或 Y 轴移动。

**Max:** 定义镜头与物体的最大距离值。

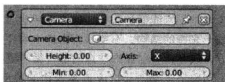


图 12-67 Camera (镜头) 促动器的属性面板

## 12.5.6 声效促动器

Sound (声效) 促动器, 用于在游戏中为一个动作或触发事件激活一个特定的声效, 如图 12-68 所示, 为 Sound 声效促动器的属性面板。

**Play Mode:** 选择载入的音效文件, 建议使用 WAV 格式的文件。声效促动器还提供多种播放模式, 例如 Play Stop 和 Loop Stop 等。

**Volume:** 音量控制。0 为静音, 1 为最大音量。

**Pitch:** 设置播放的速度。正值代表回放速度加快, 负值时回放的速度将减慢。

**3D Sound:** 开启 3D 音效。可自定义环境增益等音频特效。



图 12-68 Sound (声效) 促动器的属性面板

## 12.6 游戏开发与性能优化

经过多次对 BGE 的版本升级和功能更新, GE 内核相比早期版本已得到了极大的提高,特别是强化后的引擎运行性能。在 3D 游戏中,常使用 FPS (Frame Per Second) 指标值来评测游戏的运行性能, Blender 中的 FPS 默认值设置为 60。当游戏运行在不同硬件的计算机或操作系统中时,性能上可能会有不同程度的差异,同一个游戏在某一台电脑上能达到 60fps,但是在另一台中可能只能达到 20fps。因此,为了保证游戏的流畅运行,在游戏的开发过程中,必须十分注重对性能的处理。接下来简单讨论一下在 Blender 中开发游戏时可能使用到的一些优化技巧。

### 12.6.1 建模

从严格意义上说,游戏采用的三角形模型能较快地提升引擎的计算速度。因为游戏引擎在读取网格后, CPU 将只计算三角形的数量,所以即使使用了四边形模型,运行中单个四边形也会被转换为两个三角形来参与计算,这就增加了双倍的运算负担。如图 12-69 所示,为使用 Blender 参与模型制作的《Torchlight》截图。

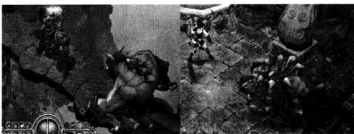


图 12-69 《Torchlight》截图

据测试, Blender 能承受一个 500 000 左右网格面的游戏场景。在建模的过程中,应尽可能地减少模型上点线面的数量,建议不要尝试通过细分网格来为模型添加细节。同时也没有必要创建过于完整的结构,不要保留物体上看不到的面,将这些面省下来并用可以在丰富显眼位置的地方,或者把容易让人注意的部位做得更精细点,其余区域稍微简化。比如在游戏中,玩家在视觉上印象最深刻的部位应该是人物的脸部,因此可以把更多的多边形用在角色的脸部结构上,而对于手部和腿部等容易被忽视的区域,就没必要做得过于精细。

### 12.6.2 材质与贴图

如果材质的某一个属性值被设置为 0.001 或者 0.98,那么它们的实现效果和 0 或 1 没有任何区别,但是这个差值却会为 Blender 带来更多的运算量,特别是 Shade (着色) 计算。在 GLSL 着色运算中, Lambert 和 Phong 的运算速度是最快的,所以应尽量选择使用这两种类型的着色器。高光着色器的默认设置为 Cook Torr, 完全可以使用 Phong 来替代,因为它们两者的效果几乎是一样的,如果没有必要使用,甚至可以尝试关闭高光着色。

对于贴图纹理,应尽量减少 Alpha 贴图的使用数量,如果贴图时不需要应用 Alpha 透明通道,那么在纹理面板中可将这个通道的运算关掉掉,以减少多余渲染的 CPU 开销,实现降低计算量的优化。同时还应尽量注意纹理渐变的混合模式,建议使用在运算速度上更快的 Mix、Add、Subtract 和 Multiply 模式,避免使用 Ramp 和 Blend 模式。

在制作纹理图片时,应尝试使用较小尺寸的纹理文件,以减小系统载入纹理时的内存消耗,尺寸最好是 2 的 n 次方,例如 512 × 512。

在表现物体细节时,应尽量多地尝试使用法向贴图,利用法向运算来丰富模型的细节,而不

是使用网格模型。同理，高光贴图也可直接替代着色器，完成物体表面的实时高光效果。如图 12-70 所示，为波兰艺术家 Pawet Lyczkowski 在低模上完成高精度细节的效果，他分别使用了色彩贴图、高光贴图和法向贴图，在 1454 个面的模型基础上，制作出了惊人的细节表现效果。

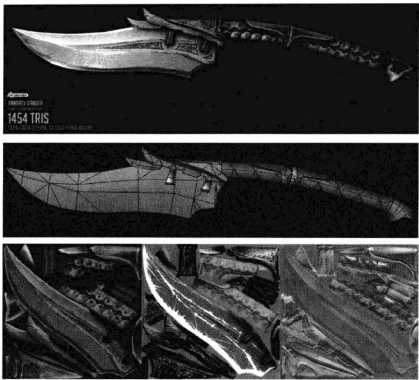


图 12-70 Pawet Lyczkowski—Fantasy Dagger

### 12.6.3 灯光

由于游戏中的实时光影计算会消耗大量的资源，所以在制作环境物体时应尽量利用阴影烘焙，以及 AO 烘焙。对于场景中人物的照明，则可以利用图层来实现对单独对象的照明，而不要尝试去照亮整个场景。

同时，还应尽量调低阴影的缓冲值，利用 Only Shadow 属性的灯光做实时的阴影渲染，并关闭灯光的高光照明，这样就可以将资源节省下来，分配给其他运算，例如同场景中的粒子特效运算等。如图 12-71 所示，为 Blender 开源游戏《Yo Frankie!》中，对场景里的树木和石头执行阴影烘焙的效果。

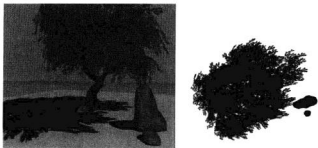


图 12-71 《Yo Frankie!》对场景中树木和石头阴影的烘焙效果

## 12.6.4 物理与逻辑

对于物体的 Physics（物理）属性设置，如果类型为 Dynamic 的物体过多，游戏的运行会变得令人惊讶的慢。同时在重力设计中，也应尽量使用合适的重力值，如果重力设置过大将有可能使 GE 的运行结果超乎预料。

在 Logic（逻辑）编辑中，还应尽可能少用 Collision 和 Touch 类的触发器，它们将极大地降低游戏的运算速度，可以通过添加一个不可见的 Collision 面来完成同样的效果。同时，对于一些无需碰撞计算的面或物体，一定要去掉其 Collision 属性。要多利用 Add Background Scene 和 Add Overlay Scene 两种场景促动器，因为合理地场景分配可以有助于避免大量的麻烦。

## 12.6.5 Python 程序代码

在 Python 代码设计中，尽量少使用 Drot 和 Dloc 参数来调用运动促动器，因为它们是按帧计算的。也就是说，同样的游戏在性能好的电脑上运行得很快，而在性能较差的电脑上运行会加倍的慢。

为了保持兼容性，应尝试使用代码来完成部分属性和参数的设置。例如物体的默认接触距离为 0.04，但是如果将数值调得过小，将会使在地面上运动的物体变得十分不稳定。所以可以使用如下代码来完成属性的重置：

```
controller = GameLogic.getCurrentController()
owner = controller.getOwner()
owner.setCollisionMargin(0.1) #将属性值重置为 0.1
```

记住，系统提供的结点永远无法满足逻辑设计需求，同时也不一定是性能最优化的制作方案。再复杂的逻辑功能设计都可以用 Python 程序来简化完成，而同样的逻辑功能，使用 Python 代码一般都会比结点的运行性能更快。所以如果是想学习游戏开发的读者，请一定要多去熟悉 BGE 的 Python 文档。

## Part3 第三部分

# 实 例 篇

在接下来的4章实例教程中，主要讲解如何使用 Blender 来完成一套完整的项目制作。其中，第13章将重点介绍模型的常用布线思路，以及建模中需要注意的制作技巧；第14章以一个人像模型为例，重点讲解了雕刻、材质与贴图、后期合成等静态制作中的高级应用；第15章将深入了解 Yafaray 这款强大的产品渲染器，以及制作室内效果图的基本流程；最后在第16章中，会简单创建一个角色模型，并详细讲解如何制作动画人物的骨骼，以及相关的约束设计和动作制作。

第13章 汽车建模与机械绑定

第14章 角色模型制作

第15章 室内效果图应用

第16章 角色动画





## 汽车建模与机械绑定

在本章中，将学习一套完整的汽车建模工作流程，主要讲解如何使用 Blender 来完成模型的制作，以及一些基本的机械约束绑定设计。本章将着重讲解建模中的常用技巧，会从表面布局思路、建模顺序、硬边处理、修改器的复用、约束绑定等几个方面来介绍这个模型的制作，同时简单介绍室内照明模式。

### 13.1 准备与分析

#### 13.1.1 添加背景参考图

即使不需要做到工业级别的精确尺寸，大量的参考图片也是一个高质量模型的基础，完美的细节处理会让模型看上去更引人注目。可以从网络上找到大量的高清参考图片，建议初学者准备一份标准的设计三视图，这可以有助于在建模的过程中把握整体的比例与局部的结构。

在 Gimp 或 Photoshop 中打开奥迪 R8 三视图文件，使用裁剪工具，将每一部分都剪切为单独的图片，如图 13-1 所示。需要分离出前视图、后视图、侧视图和顶视图，并分别命名存储。

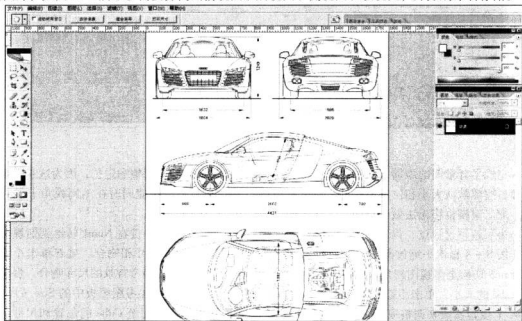


图 13-1 打开三视图

启动 Blender，首先设置背景参考图。在视图窗口中单击小键盘的 Num3 进入侧视图视角，然后使用快捷键 N，激活面板中的 Background Images（背景图）选项，如图 13-2 左图所示。接下来单击下面的 Add Image（添加图片）按钮，并在弹出窗口中选择准备好的侧视图文件，再将 Axis 约束坐标设置为 Right，这样当前参考图就会被固定显示在侧视图中，如图 13-2 右图所示。

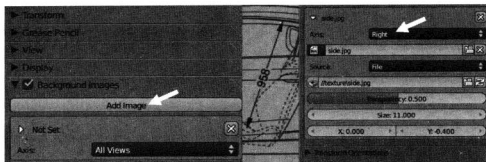


图 13-2 锁定参考图

接下来，建立剩下的前视图、后视图和顶视图等背景参考图。在视图窗口中单击小键盘的 Num1 进入前视图视角，使用同样的方法选择并添加前视图图片，并将 Axis 锁定为 Front。同理，使用小键盘 Num7 进入顶视图视角，建立顶视参考图，锁定 Axis 为 Top。单击 Ctrl + Num1 进入后视图视角，建立后视图参考图，锁定 Axis 为 Back。通过这些操作，就可以在同一个窗口中的不同视角下使用对应的参考背景图了，如图 13-3 所示。

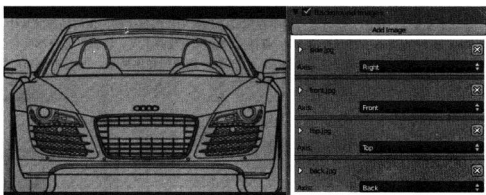


图 13-3 单窗口中的多参考图设置

完成了背景图的添加，还需要对参考图之间的相对位置做一些调整和定位，因为这 4 张参考图都是按照默认大小和位置放置的，所以物体的位置和尺寸不一定能同时在不同视角上做到相互匹配，建模也将无法做到准确定位。

为了调整这一点，首先单击 Shift + A 添加一个 Cube 物体，使用小键盘 Num1 转换到前视图视角，按 S + X 做水平缩放操作，使 Cube 的宽度与前视图的背景图尺寸相吻合。接着单击小键盘的 Num3 切换至侧视图视角，使用 S + Y 让 Cube 的长宽与侧视图的参考背景图尺寸吻合。保持此时 Cube 的大小，单击小键盘的 Num7 切换至顶视图视角，通过修改参考图面板中的 Size（尺寸）和 X-Y 位置参数来调整顶视图的背景图，如图 13-4 所示，将它匹配至 Cube 的位置和尺寸。同理，也如此调节后视图的参考图位置。

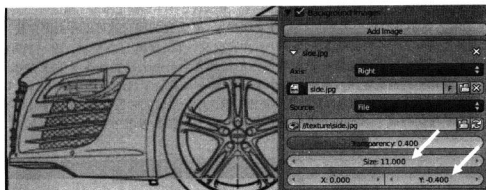


图 13-4 参考背景图间的相对定位

通过这些调整，相对定位就完成了。此时当在 3D 环境中旋转视图时，零件的位置就会在不同视图中被准确地定位出来，方便了后面的建模工作。

### 13.1.2 分析建模思路

在开始一切建模工作之前，最重要的事情就是先分析这些参考图，理清建模的思路。初学者切忌凭空就动手直接开始制作模型，应该适当准备一些布线草稿图，有助于有序思考，并设计出合理的布线结构。

机械模型的布线法则是让主体线条跟随着零件的布局方向走，并且尽量在边缘部分使用 Loop（循环）结构。同时在面之间的连接处尽量使用四星点，避免出现三星点和五星点等特殊结构。一个合理的布线结构可以便于制作硬边和倒角效果，同时也能使模型表面在细分后看上去更平滑。

首先来了解一下 Loop（循环）结构的概念。Loop 结构是一种循环的多边形分布结构，当对一个多边形表面使用快捷键 Ctrl + R 做 Loop Cut（循环边界切割）时，非 Loop 结构的物体切割线将受到非四边形的边界影响而终止，如图 13-5 左图物体所示。而具备 Loop 结构的模型能产生一个循环的切割线，如图 13-5 右图物体所示。这样的结构既能便于制作更多的细分细节，而且还可以减少边界位置处受非四边形影响产生的多余交点。可以使用挤压的方式来制作 Loop 循环结构，也可以通过对面的重缝合操作来调整多边形的表面布局。Loop（循环）布线方法是在接下来的建模过程中使用最广泛的技术之一，也是必须要理解和掌握的一种布线技巧。

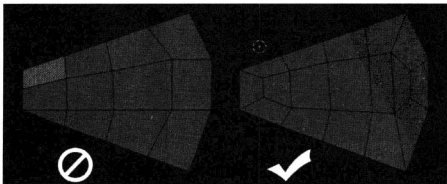


图 13-5 Loop 循环结构的布线规则

以车头部分为例，在侧视图中单击快捷键 N 打开面板中的 Grease Pencil（画线）工具，激活 Cursor（光标）选项并添加一个 New Layer（新图层）。按住 D 的同时就可以在视图中使用 LMB 直接绘制参考线条了，当然也可以在 Photoshop 中完成这一步。如图 13-6 所示，为一个简单绘制的布线思路图，其中所有的布线都紧贴着零件的纹理方向。在引擎盖和车身的接缝处使用了 Loop 布局，见图中的箭头方向，车轮为一个半圆形 Loop 结构，车灯的位置也采用一个 Loop 循环。可以在不显眼的位置使用非四星节点，见图中间箭头的位置。同时还需要注意一点，零件之间的大缝隙将车体切分成了多个单独的部分，所以一定不要在参考图中将一条线从车头一直画到尾。

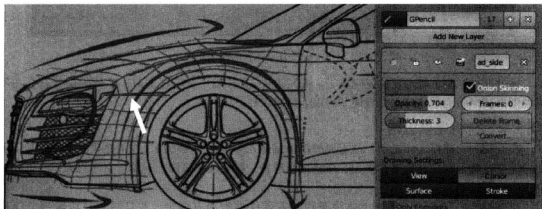


图 13-6 线条布局思路

## 13.2 车头建模

### 13.2.1 使用镜像修改器代替对称建模

接下来就正式开始建模了。首先为刚才的 Cube 添加一个 Mirror（镜像）修改器，单击 Modifiers（修改器）面板中的 Add Modifier（添加修改器）按钮，在下拉菜单中选择 Mirror（镜像）修改器，设置 Axis 镜像坐标为 X 轴方向，如图 13-7 所示。切换至视图窗口中，使用 LMB 选择 Cube 的负 X 轴侧面上的点，按 G 向 X 正方向平移，可以发现镜像修改器自动复制了车身的右半部分，这样，只需要制作车身的左半部分即可。

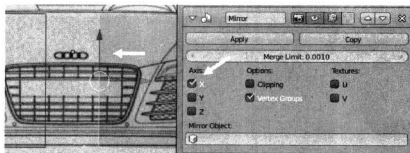


图 13-7 添加镜像修改器

在面模式下删除中间的这个面，然后激活 Mirror 修改器中的 Clipping（边界）选项，如图 13-8 所示，这样修改器会自动缝合镜像接口处的面。



图 13-8 使用 Clipping (自动缝合) 接口

### 13.2.2 制作车头部分

单击 Num3 进入侧视图视角, 可以先关闭线稿显示。单击快捷键  $\text{Ctrl} + \text{R}$  对车身做环形切割操作, 并将切割线调整至车头和车门的接缝处位置, 以便于根据这条边将车头部分和车身部分切割出来, 分别进行独立制作, 如图 13-9 所示。

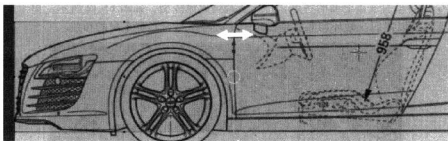


图 13-9 切割车身

在线选择模式下按住  $\text{Alt} + \text{RMB}$  选择刚切开的 Loop 线条, 然后使用组合键  $\text{Y} + \text{G} + \text{Y}$  做分离操作, 将车头和车身部分完全分离开。接着使用 C 框选 Cube 前部的点, 再按 G 将其移动至车引擎前盖的位置, 如图 13-10 所示。

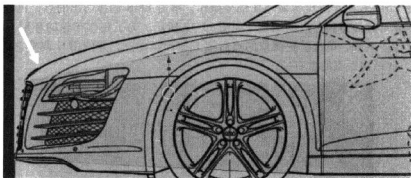


图 13-10 初步调节前盖的位置

这时可以打开线稿显示, 并分别在 Y 方向和 Z 方向上使用  $\text{Ctrl} + \text{R}$  对车头做两次环切操作, 然后删除掉车轮下面的边, 将剩下的点使用 G 进行移动操作, 把它们匹配至线稿位置上, 如图 13-11 所示。

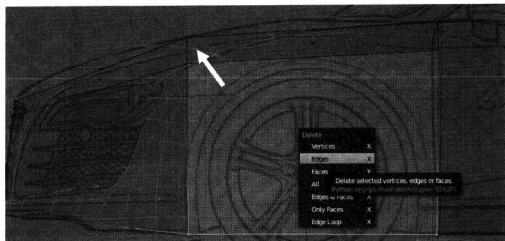


图 13-11 删掉车轮下面的边

使用 G 移动车引擎盖上的各点，将线条调整成流线型，如图 13-12 所示。由于车头已经和车体分开了，所以我们在做环切和移动操作时不必考虑是否影响车身的布线。

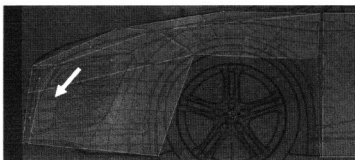


图 13-12 利用环切进行细分操作

继续使用 Ctrl + R 对车灯位置做环切细分操作，单击 RMB 选择切出的点，再按 G 移动至线稿位置，并删除部分面得到车灯的结构，如图 13-13 所示。为了制作车轮的环形 Loop 结构，可以先适当地删掉一些线，单选前轮部位的点后按 E 挤出新的点，并使用 G 调整它们的位置，为制作车轮的 Loop 结构做点的重布局。

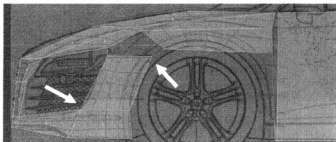


图 13-13 继续细分车灯部分

由于已经在参考图中绘制好了布线图，所以在处理 Loop 循环结构时会比较容易。继续

对表面执行环切细分操作，选择4个相邻的点然后单击F就可以缝合缺失的面，效果如图13-14所示。注意箭头处的五星点，应该尽量将这种点放置在不显眼的硬边位置处，保证它不在需要平滑的弯曲表面上，因为五星点在细分修改器的处理下会产生影响光滑的凸起面效果。

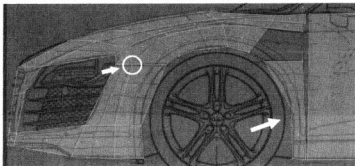


图 13-14 缝合缺失的面

按住MMB旋转视图至自由视角，由于刚才的操作都基于侧视图，所以调节的点都处在同一个平面上。此时可以看到，模型的侧面大体布局结构已经出来了，接下来需要做的就是调整它们的横向分布，让这些面产生立体感，单击Alt+R选择图13-15中所示的边。

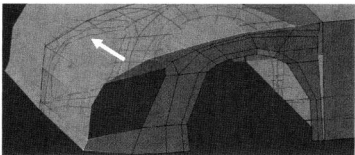


图 13-15 选择引擎盖上的边

按小键盘的Num1进入正视图视角，使用快捷键G+X在水平方向上移动这组点。将这些点放置在引擎盖的硬边位置附近，然后使用R+Z进行原地Z轴方向的旋转，如图13-16所示。

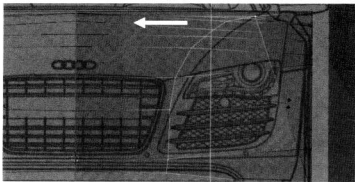


图 13-16 在正视图中做Z轴约束下的旋转调节

由于已经在侧视图中对准了点的 Y 轴坐标，所以现在只需要调整它们在 X 轴上的位置即可。使用 RMB 选择这些点后，单击 G + X 做水平移动，最后的调节效果如图 13-17 所示。

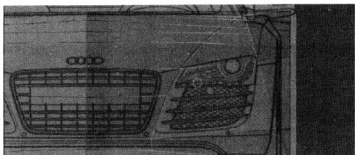


图 13-17 点的微调

同理，调整剩下的边，分别将它们平移至正视图中参考线的位置。按照刚才的方法，选择点后使用 R + Z 做原地旋转，如图 13-18 左图所示，再对点进行微调，最后的效果如图 13-18 右图所示。

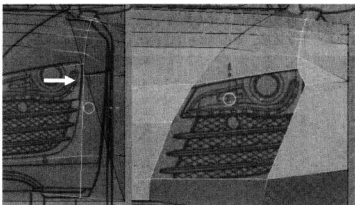


图 13-18 利用同样的方法调整剩下的点

全部调整完成后的效果应该如图 13-19 所示，这样车头的整体形态就基本上已经完成了。接下来就需要为其添加更多的线条来丰富细节。

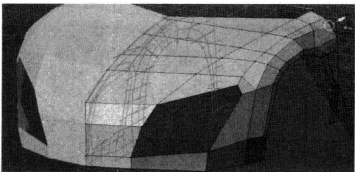


图 13-19 完成初步调节后的车头形态

按小键盘的 Num1 回到正视图视角，使用环切工具 Ctrl + R 进行多次纵向细分，并分别单选



各点进行位置微调,如图 13-20 所示。由于引擎盖具有一定的弧度,所以在微调的过程中不一定使用 G+X 工具约束水平的移动,可以适当地对点在 Z 方向上做轻微的移动。

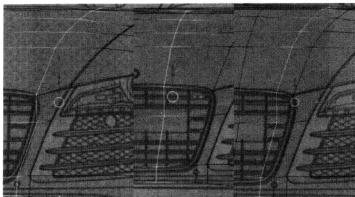


图 13-20 使用细分操作为模型添加更多的细节

不要忘了还有顶视图参考图,只有在三个视图角度上都完成对点的位置调节,才能保证模型的准确性。单击小键盘的 Num7 切换至顶视图视角,如图 13-21 左图所示,表面线条很凌乱,相邻面的面积差距较大,模型在细分后将会由于表面密度不均匀而显得不光滑。使用 RMB 单选各个点再按 G+Y 进行微调,使线条具有一定的曲率,每个面的面积尽量均匀,最后的效果如图 13-21 右图所示。

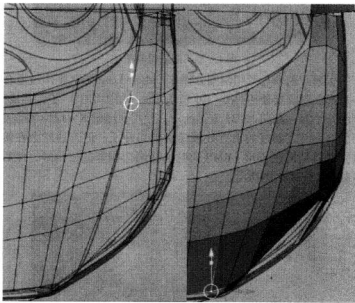


图 13-21 在顶视图中调整布线

回到侧视图视角,使用 RMB 选择并按 X 删掉前进气孔的面,在大灯的 Loop 结构上进行环切细分操作。这时会发现,基于之前布局的 Loop 结构,细分操作将变得十分轻松,切割点分布也自然有序,效果如图 13-22 所示。现在只需要先做环切然后再微调,而不必担心点的分布问题。由此可以看出,一个好的布线习惯是影响建模效率的重要因素之一。

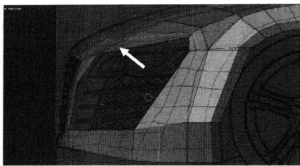


图 13-22 在 Loop 结构上进行细分操作

继续做环切细分，记住每一次操作后都需要在三个视图中同时调整，这样才能保证模型的精确性，如图 13-23 所示。

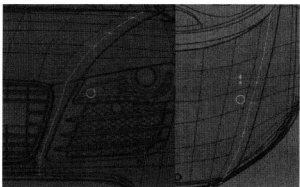


图 13-23 细分引擎盖

接下来要对车的引擎盖前端进行切割细分操作。由于 Loop 切割操作最适合于大面积的 Loop 结构，而对于一些局部面就需要使用另外一种 Knife Cut（精确切割）了，其快捷键为 K。在面模式下选择要被切割的几个面，如图 13-24 上图所示。使用快捷键 K + LMB 在需要加线的位置处划出切割线，一个面就会在刚才划过的位置处被精确地切开了。但是这样会在接口处产生一些多余的三角面，解决方法就是按住 Shift + RMB 同时选择两个相邻的三角形，然后单击 F 将它们合并成一个四边形，如图 13-24 下图所示。

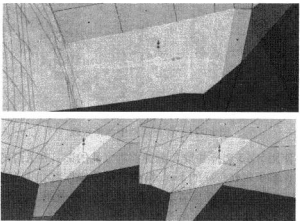


图 13-24 精确切割面

曲线总能赋予物体优美的感觉,所以即使还没有对物体做进一步的细分操作,也要尽量在低模的情况下制作出线条的弧度感。单击小键盘的 Num1 切换至正视图视角,如图 13-25 左图所示,继续使用环切工具 Ctrl + R,对进气口旁的的面做多次细分。接着再单击 Num3 进入侧视图图进行位置微调,平移这些点制作出线条的弧度来,如图 13-25 右图所示。

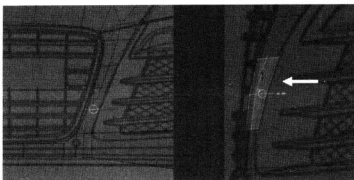


图 13-25 细分车前面

前面曾讲到,建模的过程中应尽量控制各相邻面的面积过渡均匀,这样,表面在细分后(特别是经过多次细分后),网格才能变得十分平滑。继续使用 Ctrl + R 工具添加切割线,不断地切换视图并调节点和线条的位置,如图 13-26 所示。

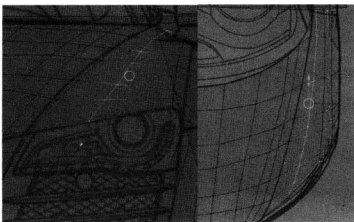


图 13-26 添加切割线

线稿图上还有一些线条没有添加,所以可以放大视图,继续使用 Ctrl + R 对模型做环切细分操作。这里没有太多的技巧,只需要遵照线稿布局做就可以了,如图 13-27 所示。

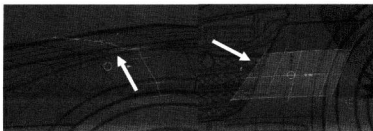


图 13-27 继续做环切操作

保险杠的位置需要一个凹槽硬边, 由于在这里刚好有个 Loop 结构, 所以可以在 Loop 面上进行环切操作, 如图 13-28 所示。

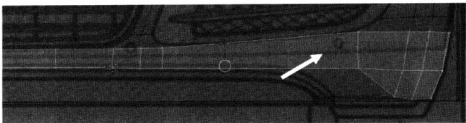


图 13-28 制作保险杠凹槽硬边

在经过两次环切操作后, 开始调节点的位置。从参考图的侧视图中可以看到, 保险杠的边缘弧度并没有贯通车头, 而是在距离车轮一定距离的位置融合进了车头, 那么这时可以考虑使用三角面来制作这个转角。按住 Shift + RMB 先选择上下 2 个点, 再选择中间的点, 也就是打算最后合并的点位置, 然后单击快捷键 Alt + M 使用 Merge (合并) 功能, 如图 13-29 上图所示。选择合并到最后一个点位置模式, 将 3 个点合并起来, 得到的效果如图 13-29 下图所示。

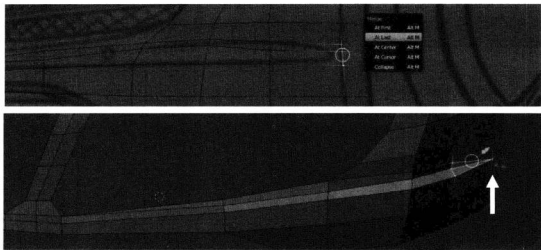


图 13-29 合并保险杠上的交汇点

继续细化细节, 并且变换视角进行微调, 如图 13-30 所示。这也许是最枯燥但也是最重要的步骤, 耐心地调节每一个点, 一定要保证整体的光滑, 不要忘记把所有的线稿都画出来。

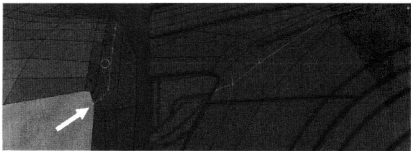


图 13-30 细化车头引擎盖细节

## 13.3 车身建模

### 13.3.1 车身布线思路

接下来单击小键盘的 Num3 进入侧视图视角, 新建一个绘制图层, 单击 D + LMB 开始绘制车身部分的参考布线图。R8 的车体拥有十分经典而完美的整体流线形, 因此在布线的时候, 应尽量保持主线条流畅贯通的风格。具体的布线思路应该从整体出发, 以主要的接缝为界, 在上面逐渐细分, 同时注意保持线条的疏密和流畅感。

首先画出车顶和车门的结构, 如图 13-31 上图所示。后车轮处需要设置为 Loop (循环) 结构, 同时还需要注意后轮区域与车门上线条走向的转折关系, 如图 13-31 中图所示。车门后侧的通风口与车门是分离的, 所以不需要考虑这里的连接线条, 最后的效果如图 13-31 下图所示。

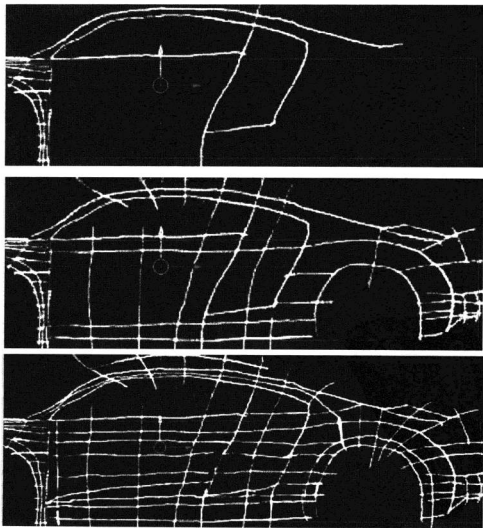


图 13-31 绘制车体后半部分的布线思路图

### 13.3.2 车身建模

开始根据布线图对车身建模，首先按照前面讲的分割技术对车体进行切割操作，将分割边移动至车门的接缝位置，如图 13-32 所示。

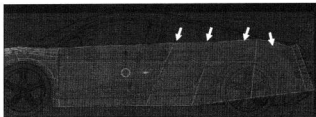


图 13-32 对车身做切割操作

删掉车轮位置的线条，然后单击 RMB 选择旁边的点，按 E 做挤压操作，使用 G 调整点的位置，待确定位置后选择相邻的 4 个点按 F 做面缝合操作，接着再使用 Ctrl + R 进一步细分。车体线和车轮 Loop 区域的地方先暂时保留，需要使用一些方法来单独处理这个五星点，如图 13-33 所示。



图 13-33 制作后轮上的 Loop 结构

首先在车门位置进行切割，然后采用挤压点的方式连接放置五星点的位置，见图 13-34 圆圈处。在该点的周围再进行面缝合，一切顺利的话会得到如图 13-34 所示的效果。

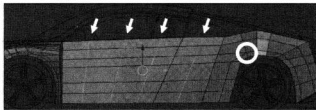


图 13-34 环切细分后的五星点

由于环切的时候线条是根据被切割线的两端距离进行平分的，因此可能会遇到一些杂乱的线条。为了调节它们的排列状态，可使用 Alt + RMB 进行 Loop 选择，然后单击 S + Y + Num0，将这些点在 Y 方向上自动对齐，效果如图 13-35 所示。

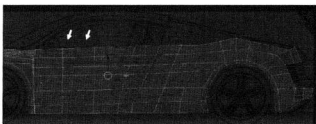


图 13-35 调整后的车体布线

使用 MMB 切换至自由视图视角，会发现车身在调整细分时，除了车门顶部边缘上的点能和

车头对齐，其他点还具有一定的差距。为了能精确地连接车身和车头，在这个地方需要激活吸附功能，单击状态栏中的磁铁图标，如图 13-36 所示，并选择为吸附至最近的点位置模式。



图 13-36 启用点的吸附功能

接着使用 RMB 选择需要被调整的点，单击 G 向目标点位置移动，吸附功能会自动将这两点重合起来，如图 13-37 左图所示。松开鼠标，因为车门和车头之间还需要保留一定的接缝，所以再次单击选择这个点，按住 G + Y 向正轴方向做少许移动，如图 13-37 中图所示。这样不仅完美地连接了车头和车身部分，还精确地制作了接缝效果。紧接着再采用类似的吸附方式，将剩下的其他点做连接操作，如图 13-37 右图所示。完成修改后应记住关闭吸附功能。

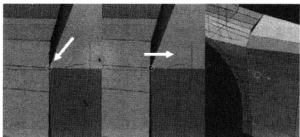


图 13-37 使用吸附功能来完成点与点之间的精确衔接

单击小键盘上的 Num7 切换至顶视图视角，然后使用 Alt + RMB 做 Loop 操作，选择车门顶部的点，对它们的位置进行微调，如图 13-38 左图所示。接着选择车门上剩余的点，在 X 轴上也做一些移动调整，让线条均匀地分布在车身上，效果如图 13-38 右图所示。

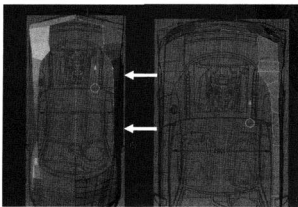


图 13-38 调整车身侧面的布线

由于点的数量较少，所以调整会很快。完成了初步的修改，就可以删掉车尾上后车灯位置的一些点，制作出车尾灯部分。接着在车身上继续做细分，沿着 Loop 结构面进行切割操作，并且在不同的视图图中进行位置调整，最后的效果如图 13-39 所示。

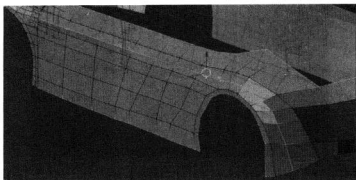


图 13-39 使用切分工具继续细分车身

车体的裙边和车身之间也有接缝，为了将它们分离，首先使用 Loop 选择的方式，选择切割位置上的 Loop 边，如图 13-40 上图所示。然后单击快捷键 V + G 在 Z 轴上稍作移动，线条就可以从接缝处被分离开了。同理，也可以将车门和车身进气口部分的接缝分离出来，使用 Loop 选择分离边，然后按 V + G + Y 向负 Y 方向移动即可，如图 13-40 中图所示。最后再切割出车门的底部与车身衔接的线条，效果如图 13-40 下图所示。

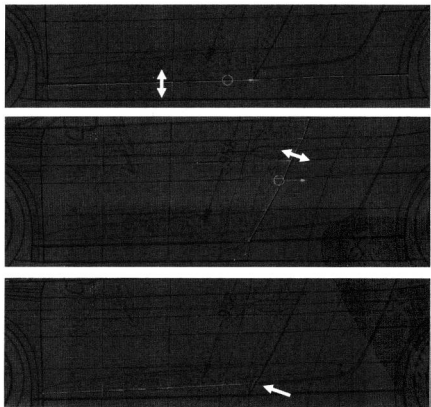


图 13-40 分离出车门与周围部件的接缝结构



接着开始调节车门上的凹槽，注意到车门凹槽的顶部线条刚好处于门把手边缘的位置，凹槽下方的线条在车门底部形成了一个弧线形。所以紧贴这两个区域做环切操作，如图 13-41 上图所示，并使用 Alt + M 合并凹槽线前端与车门上的点，制作出渐进线效果，如图 13-41 中图所示。最后调整出凹槽下方的流线线条，旋转视角，效果如图 13-41 下图所示。

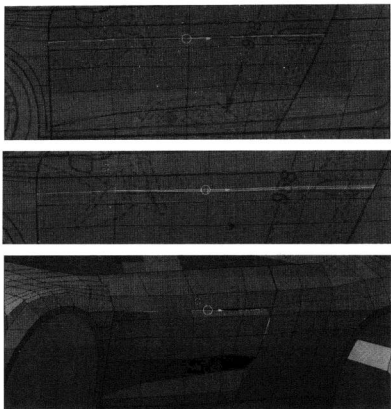


图 13-41 制作车门凹槽

紧接着，使用 Shift + RMB 将车门中部的几个点同时选中，使用组合快捷键 G + X 向负 X 轴方向移动，将整个面向门内挤压制作出凹槽，如图 13-42 左图所示。同理，需要将门中间的其余点都向门内移动，最后得到的车门凹槽效果如图 13-42 右图所示。在编辑的过程中，可以使用 Alt + B + LMB 框选显示想要修改的部分，将其他无关的区域都隐藏起来，这在查看复杂模型时十分方便灵活。完成编辑后，只需要再次单击 Alt + B 就可取消这种视图隐藏模式。

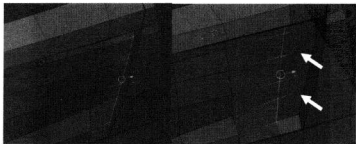


图 13-42 调整车门凹槽面上的点

完成全部的调整，应该能得到如图 13-43 的效果，这样门的进气凹槽部分就做好了。

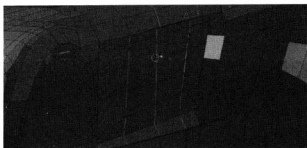


图 13-43 车门布线的最终效果

再转至车门与车身的连接位置，由于凹槽始于车头，所以选取接缝处的边，按 K + LMB 进行精确切割，如图 13-44 左图所示，调节各点的位置后如图 13-44 右图所示。



图 13-44 制作车门接缝细节

选择车顶的支柱网格部分，按 E 向上挤压边，如图 13-45 左图所示，并使用 RMB 对点进行移动微调，让模型紧贴着线稿图，如图 13-45 右图所示。



图 13-45 制作后门柱

当完成了大部分的结构制作，就可以关掉 Grease Pencil 上层的小眼睛按钮来取消线稿图的显示，使画面更干净，方便观察和调整。车门框在连续多次挤压后可以得到如图 13-46 所示的效果，在前视图中，对这部分点再做一定的调整。选择门框中间的 Loop 线稍微往外移动，这样门框就会呈现轻微的凸起效果。



图 13-46 门框最终效果

接着来制做车顶，单击小键盘的 Num7 进入顶视图视角，使用 Shift + RMB 复合多选车顶上的边缘点，然后使用快捷键 E + G + X 向负 X 轴移动，如图 13-47 左图所示。由于开启了 Mirror（镜像）修改器的 Clipping 功能，所以在接缝处的面会被自动缝合，如图 13-47 右图所示。

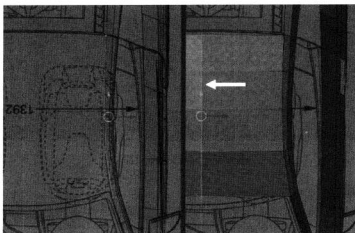


图 13-47 制作车顶

单击小键盘的 Num1 进入前视图视角，使用 Ctrl + R 环切工具对车顶做细分操作，并使用 G 在 Z 轴方向上做微调操作，如图 13-48 所示。

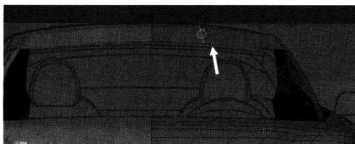


图 13-48 调整车顶

同理，使用前面制作车顶的操作方法来制作车尾部分，首先使用 E 将临界边挤压至画面中央位置，如图 13-49 左图所示。接着使用 Ctrl + R 对该面做细分操作，然后用 G + Y 进行点的移动微调，让线条的边缘贴合三视图，如图 13-49 右图所示。

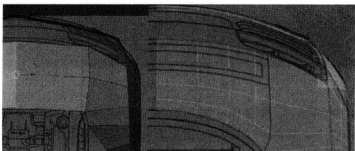


图 13-49 制作车尾

接着按小键盘的 Ctrl + Num1 进入后视图, 使用 Shift + RMB 选择多点, 如图 13-50 左图所示。然后按 E + G + Z 向 Z 轴负方向移动, 如图 13-50 右图所示。

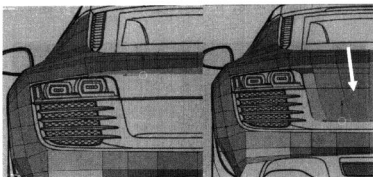


图 13-50 制作后备箱部分

### 13.3.3 车窗建模

在完成了窗框的制作, 就可以开始制作车窗了。单击 Alt + RMB 框选门窗的 Loop 线条, 然后按 Shift + D 复制这些点, 再单击 S 稍做一定的放大, 如图 13-51 所示。

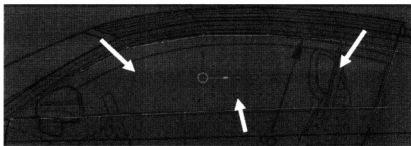


图 13-51 制作车窗

接着使用 RMB 选择面上的对角点, 按 F 将这些点连接起来, 接着使用 W 对线段做 2 次细分操作, 最后按 F 将各相邻面封闭起来, 最后得到如图 13-52 所示效果。

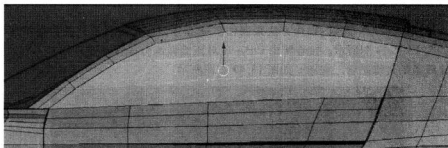


图 13-52 缝合车窗

运用制作侧门车窗的技术, 使用 Loop 选择接口线, 按住 Shift + D 进行复制, 然后单击 F 进行面缝合来制作前后车窗, 如图 13-53 所示。由于主要讨论汽车建模的思路和方法, 所以车的尾灯就不做了, 感兴趣的读者可以自己尝试将这一部分给补上。

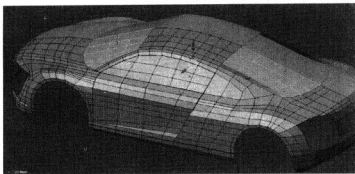


图 13-53 制作剩余车窗

为了得到更平滑的效果，将车头和车身看做一个整体，使用环切工具来继续做细分操作，效果如图 13-54 所示。可以看到最后的 Loop 线分布很均匀，线条自然地贯通车体。车身侧面上的一条主线从前车灯一直延伸到尾灯，而这也作为 R8 的主要硬边线，用于表现整体的流线形。

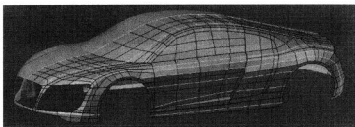


图 13-54 车体上的 Loop 布线图

针对车身的接缝部位，可以结合 Loop 选择工具和 V 分离操作，完成所有部件间的边分离。为了方便演示，这里使用了不同的颜色来标识车体的主要独立组成部件，如图 13-55 所示。这样一个简单的车体表面就做好了，接下来要做的就是向车体内部添加更多的细节。

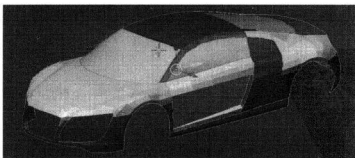


图 13-55 将部件分别标识

## 13.4 附件建模

### 13.4.1 挤压厚度

真实世界中的车体不可能只有一层皮，物体的外壳都是有一定厚度的，在 CG 制作中也应该还原这一特性，为车体添加一定的厚度，同时，一定的厚度也能方便制作车体的硬边效果。以前

保险杠为例，如图 13-56 所示，单击 Shift + Alt + RMB 对车体的边进行 Loop 选择。

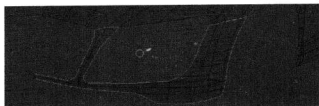


图 13-56 选择车头的 Loop 边

单击组合键 E + S 向车的内部执行边挤压与缩放操作，在缩放的同时可以按住 Shift + Z 锁定 Z 轴为缩放的约束轴，使缩放操作只对 X 和 Y 轴有效，最后的效果如图 13-57 所示。

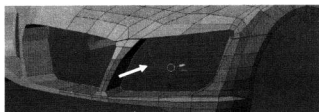


图 13-57 挤压出车体厚度

同样的，完成车体上所有部件的厚度制作。单击 Mirror 修改器上的小眼睛图标，暂时隐藏镜像修改器，可以从另一个角度来看对所有边做的厚度挤压效果，如图 13-58 所示。

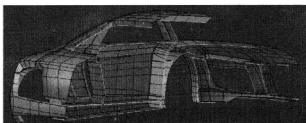


图 13-58 厚度效果

### 13.4.2 制作车头护栏

完成了车体厚度的挤压，在车头位置，单击 Shift + A 添加一个新的 Cube 物体，如图 13-59 左图所示。然后单击 S + Z 在 Z 轴上做垂直缩放操作，并删除两边的面，如图 13-59 右图所示。

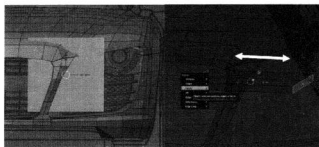


图 13-59 添加 Cube 来制作护栏

单击 Ctrl + R 在 X 方向和 Y 方向上进行多次环切操作，再使用边选择模式按住 G 将这些边在 Y 轴上移动，使 Cube 成为如图 13-60 上图所示。接下来在上面单击任何一点，按住 Ctrl + L 选择关联的物体，并使用 Shift + D 进行多次复制，同时配合 G + Z 移动操作将其放置于进气栏的垂直位置处。最后使用组合键 R + X + Num9 + Num0 将进气栏在 X 轴上进行 90 度垂直旋转，然后再次使用复制和移动操作完成剩余栅格的制作，最后的效果如图 13-60 下图所示。

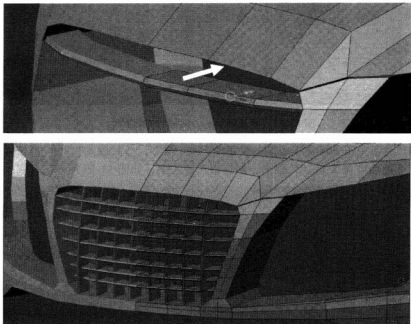


图 13-60 复制出全部进气护栏物体

运用制作车窗的方法来制作进气护栏的边缘部分，单击 Shift + RMB 连续选择车头上的边，如图 13-61 左图所示，使用 Shift + D 复制这些点，再按 S 做短距离的缩放操作。然后单击 E + S 向内挤压面，最后再使用 Ctrl + R 环切操作和 G + Y 的移动操作丰富栏杆的细节，如图 13-61 右图所示。

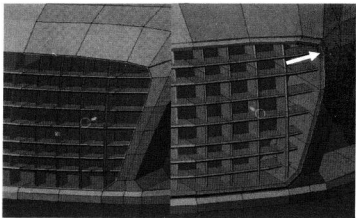


图 13-61 挤压进气护栏的边

可以使用同样的方法制作车灯的护栏，最后的效果如图 13-62 所示。

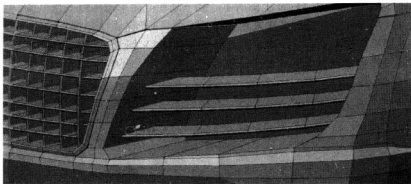


图 13-62 使用同样的方法制作车灯护栏

在车头的位置，单击 Shift + A 添加一个新的 Plane 物体，然后按 S + Shift + Y 执行缩放操作，物体的形变便会被约束在 X 轴和 Z 轴上，制作出车牌表面，如图 13-63 所示。

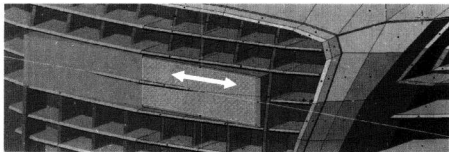


图 13-63 制作车牌

### 13.4.3 制作车灯

接下来制作车灯部分，使用制作车窗的技术复制出车灯的边缘线，然后按住 E + S 进行边的挤压和缩放操作，如图 13-64 所示。

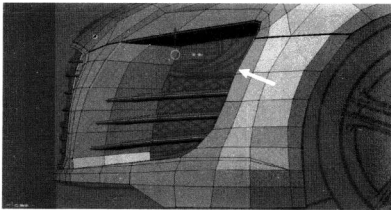


图 13-64 制作车灯

继续向内做挤压和缩放操作，就可以得到一个车灯的底座，如图 13-65 所示。



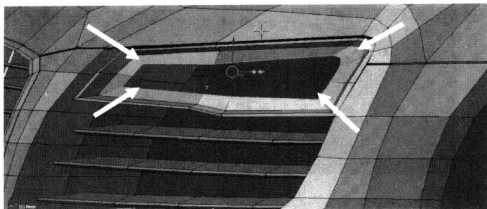


图 13-65 向内挤压出车灯的底座

根据前视图的参考图，可以通过复制点的方式挤压出一条灯管，并使用组合键 E + G + Y 向 Y 轴正方向挤压，最后的效果如图 13-66 所示。



图 13-66 新添加面来制作车灯灯管

接着单击 Shift + A 添加一个 Circle（圆）物体，将它放置在如图 13-67 左图所示的大灯位置上，保留点数量的默认设置。然后使用 E + G 操作不断挤压并在 Y 轴上移动，同时使用 S 进行相应的 X 轴缩放操作，这样就能制作出车的大灯。最后剩余的点使用 Alt + M 做合并操作，选择合并点在 Center 即可，效果如图 13-67 右图所示。

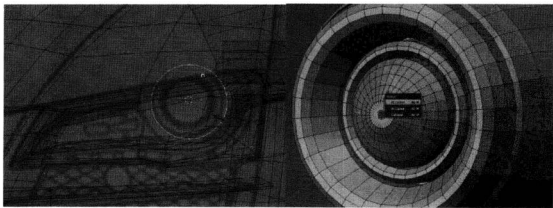


图 13-67 制作车大灯

在大灯上任意点被选择的情况下单击 Ctrl + L 做关联选择，调整它的位置，如图 13-68 所示。

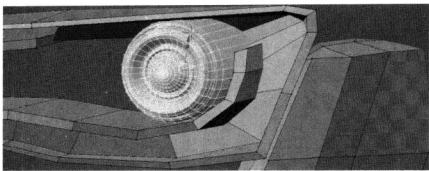


图 13-68 调整车灯

接下来单击 Shift + A 添加一个 Plane 物体，移动到如图 13-69 所示的位置，执行挤压和缩放操作来制作车内的灯管。

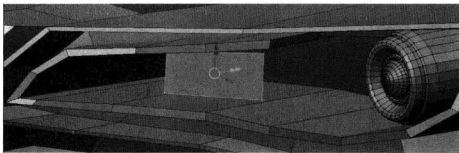


图 13-69 添加灯管细节

同时使用 Shift + D 复制车的大灯，单击 S 将其缩小，再经过 Shift + D 复制后使用快捷键 G 移动这些小灯，将其排列成一定顺序，最后的制作完成的车灯效果如图 13-70 所示。当然还可以再制作一个灯罩，将大灯内的物体保护起来，这一步留给读者自己操作。

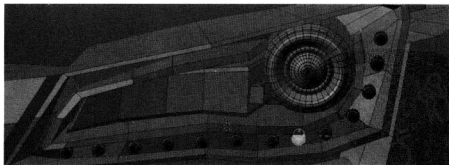


图 13-70 添加小灯泡

#### 13.4.4 制作车头网格护栏

接着再次使用制作车窗的方法复制并挤压边，完成车灯下的小护栏制作，如图 13-71 所示。

单击小键盘的 Num1 进入正视图视角，按住 Shift + A 添加一个 Plane 物体，选择面后按 W 执行 Subdivide Smooth（平滑细分）一次，如图 13-72 左图所示。切换至边选择模式，选择侧边和顶边再

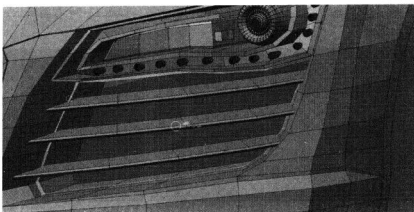


图 13-71 制作车灯下的护栏

分别单击 G + X 和 G + Y 移动成如图 13-72 中图所示。接下来切换到面模式，按住 Shift + RMB 进行多选，再按 X 删除面。激活吸附功能，将吸附模式设置为最近的边，如图 13-72 右图所示。

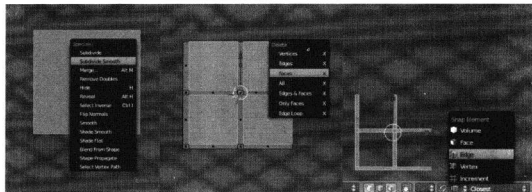


图 13-72 制作网格

使用 Shift + D + G + Y 复制当前网格单元，并向正 Y 轴方向移动，这时多边形的左侧边会自动吸附至被复制多边形的右侧边上，如图 13-73 左图所示。但是这时会出现一个位置上两个点重合的问题，解决办法是同时选择这两个物体，然后单击 W 使用 Remove Doubles（删除重合点）功能，这样网格物体间的重合点就被自动合并了，如图 13-73 右图所示。

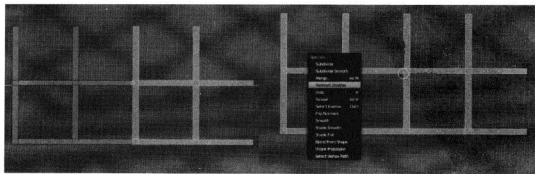


图 13-73 扩展网格并删除重合点

使用同样的方法进行多次复制、移动和重合点删除操作，最后得到了一个均匀的网格物体多边形，如图 13-74 左图所示。然后全选该物体，并使用组合键 S + Num4 + Num5 将其旋转 45

度，并单击 S+Z 做适当的缩放，比例和尺寸以三视图为准，如图 13-74 右图所示。最后删除超出网格边缘多余的面。

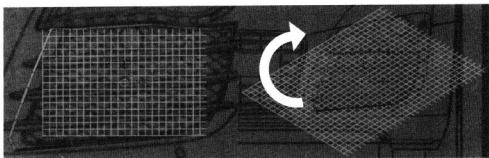


图 13-74 旋转并压缩网格

至此车头部分就基本上完成了，效果如图 13-75 所示。接下来制作车身的其余部分附件。

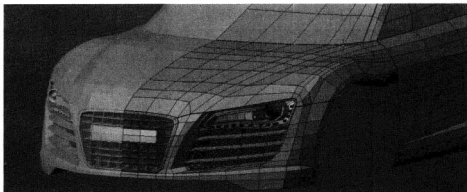


图 13-75 车头基本完成

### 13.4.5 制作车门把手

首先来制作车门把手，R8 的门把手镶嵌在车门中，所以需要在车门的面上切割出把手的形状。单击小键盘的 Num3 切换至侧视图视角，在面模式下使用鼠标 RMB 选择需要切割出的把手的面，按住组合键 K + LMB 在上面进行精确切割，如图 13-76 左图所示。然后再次选择刚才切割出来的面，按住 K + LMB 进行第二次精确切割，制作 Loop 边的凹槽线，如图 13-76 右图所示。

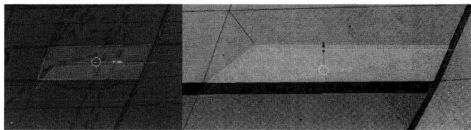


图 13-76 切割车门把手结构

面在切割后会产生一些三角面，这些面在平滑后会产生尖锐的转折边缘，如图 13-77 上图所示。可以选择两个相邻的三角面，然后使用 F 进行缝合操作。完成后使用 Alt + RMB 选择刚才切出来的 Loop 凹槽面，运用组合键 E + X 向负 X 轴方向做挤压操作，如图 13-77 下图所示。门的把手就制作好了。

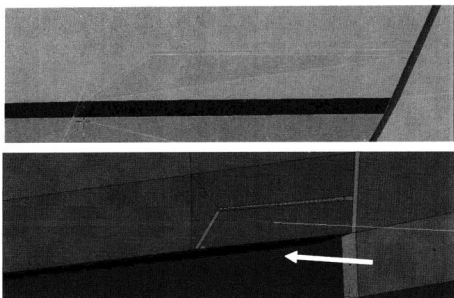


图 13-77 缝合面对把手缝隙做挤压操作

### 13.4.6 制作后视镜

将光标移动至后视镜的位置，然后单击 Shift + A 添加一个 Cube 物体，选择这个 Cube 并按 W 单击菜单中的 Subdivide Smooth（平滑细分）功能，将 Cube 做一次二次细分操作，如图 13-78 左图所示。接着按小键盘的 Num1 进入前视图视角，参考三视图，对后视镜进行点的位置微调，如图 13-78 右图所示。

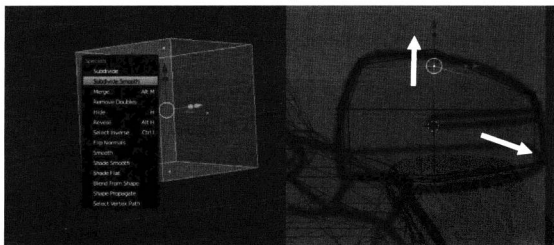


图 13-78 使用 Cube 制作后视镜

使用 Ctrl + R 对后视镜进行多次环切细分操作，然后在不同的视角中进行点位置调节，如图 13-79 左上图所示。接着使用 Shift + RMB 选择一小部分的面，单击 E 向内做挤压操作，制作出后视镜上的车灯凹槽，如图 13-79 右上图所示。最后使用 Ctrl + Num1 转向后视图，选择背面部分的面，然后按 E 向内挤压，制作车后视镜的镜面部分，如图 13-79 下图所示。

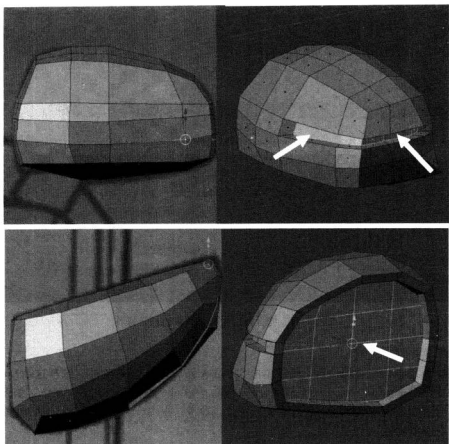


图 13-79 为后视镜添加细节

选择后视镜底座上的面，按 E 向车身方向挤压，如图 13-80 左图所示，完成后的车身与后视镜效果如图 13-80 右图所示。

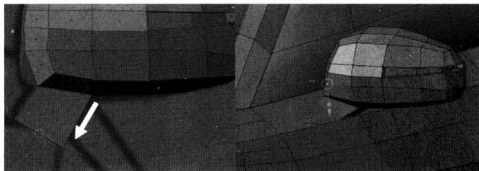


图 13-80 将后视镜装配到车身上

## 13.5 车轮建模

### 13.5.1 制作轮毂

接下来将学习如何制作车轮。首先观察车轮的参考图，如图 13-81 左图所示，R8 的车轮是

标准的五星结构，呈规则的放射状形态。同时，每一个轮毂在扇区内又是对称性结构，这样在建模时只需要制作轮毂轴的十分之一，如图 13-81 右图所示，然后利用阵列修改器和镜像修改器来自动复制剩下的部分即可。为了方便制作，可以使用 D + LMB 绘制一个简单的 Grease Pencil（线稿图），绘制布线时要注意，轮毂的布线区域一定要保持 Loop 边环绕硬边的建模原则，以及布线走线跟随物体放射性结构的原则，这样在镜像复制后物体的交界边缘才能完美地衔接在一起。

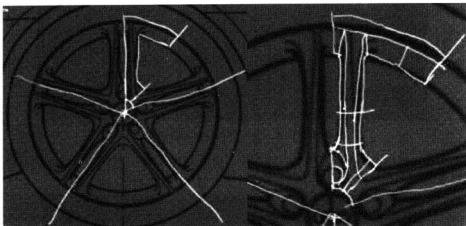


图 13-81 车轮布线思路

首先切换到侧视图视角，退出车身的编辑模式，将光标移至参考图的轮轴中心处，按住 Shift + A 新建一个 Circle（圆）物体。按照刚才的分析需要把轮毂分成 10 等分，所以在 Tool 的属性框中输入 10 即可得到如图 13-82 左图所示的 10 边圆。接着选择 8 个点，单击 X 即可删除掉其中的 9 个边，再将剩下的点全选，单击 E 向车轮中心做挤压操作，并使用环切工具 Ctrl + R 做环切细分，再配合组合键 G + X 对部分点在 X 轴上做水平移动，制作出轮毂的部分凸起形状，最后效果如图 13-82 右图所示。

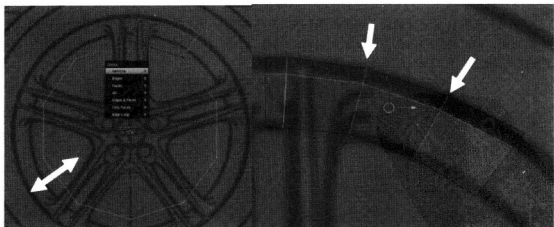


图 13-82 添加物体 Circle 制作车轮

如图 13-83 左图所示，为调整后的轮毂立体形状。切换至面选择模式，选择轮毂轴上的面，并按 E 挤压出轮毂的轴内侧部分，Blender 默认在法向方向执行挤压操作，如图 13-83 右图所示。

接着切换至侧视图视角中，将面向中心位置移动，如图 13-84 左图所示。使用边选择模式，对最外面的边做缩放操作，使轴呈现梯形形状。选择梯形的顶边，按 E 向圆心做挤压操作，并在轮毂的螺丝周围布上 Loop（循环）的边，这样可以便于在稍后的过程中挤压出螺槽，如图

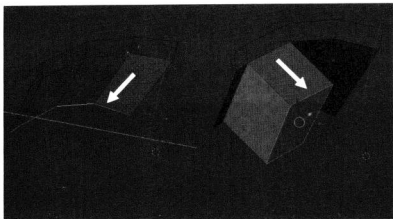


图 13-83 使用挤压制作轮毂结构

13-84中图所示。最后选择刚才挤压出来的新边，单击 E 向 X 轴负方向挤压，制作出螺槽形状，使用 MMB 旋转视图后可以看到最后的结果，如图 13-84 右图所示。

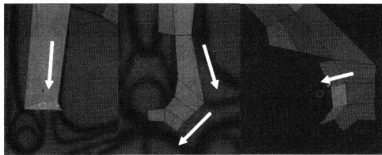


图 13-84 手动布局 Loop 结构

使用 F 来缝合未闭合的面，调节效果如图 13-85 左图所示。然后使用同样的方法挤压轮毂轴外侧的面，由于临界的边在镜像复制时会被自动缝合，所以一定要将这些位置的边处理为 Loop 四边形结构，如图 13-85 右图所示。

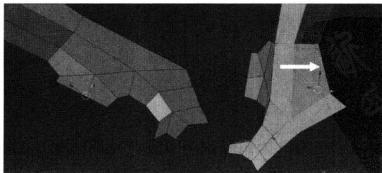


图 13-85 制作侧面的 Loop 结构

转换至前视图视角，对部分面执行框选操作，然后单击 G + X 将这些点向负 X 轴移动一段距离，制作出轴在轮毂上的弧度效果，如图 13-86 左图所示。回到侧视图视角，使用 Ctrl + R 对轴进行环切细分操作，并微调位置，由于事先布置好了 Loop 结构，所以细分操作变得十分轻松，效果如图 13-86 中图所示。物体在挤压的过程中产生了一些多余面，由于这些面均处于不可见的位置，出于优化模型



的考虑可以将它们都删掉。选择接口处的这些面，然后单击 X 删除掉，如图 13-86 右图所示。

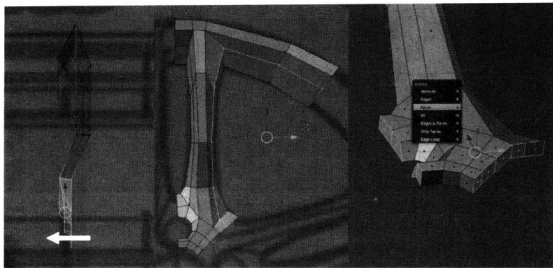


图 13-86 删除接口处的多余面

### 13.5.2 利用修改器制作轮毂镜像

接下来在修改器面板中单击 Add Modifier (添加修改器)，选择 Mirror (镜像) 修改器，将 Axis 镜像轴选择为 Y 轴，如图 13-87 左图所示。由于模型在制作中多少会产生一些误差，放大视图后可以看到，螺丝口的位置并没有完美地衔接起来。使用 RMB 选择中间的临界点，单击 G + Y 向镜像轴中心移动，修改器会自动进行面的缝合操作，如图 13-87 右图所示。

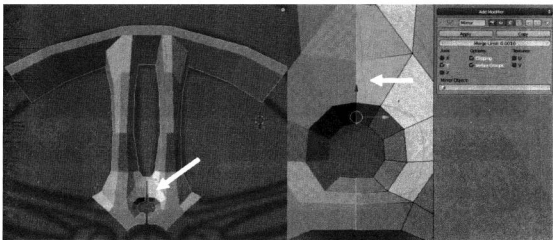


图 13-87 添加镜像修改器并调节接缝部位

为了确保在旋转复制时，新的轮毂复制体圆心正好处于轮毂参考图中的圆心位置，首先退出编辑模式，按住 Shift + S 调出 Snap 的菜单，选择 Cursor to Selected (将光标重新定位至所选物体的圆心处)，如图 13-88 左图所示。接着再使用 Shift + A 添加一个 Empty 物体，如图 13-88 右图所示，并单击快捷键 N 在属性菜单中将其命名为“tie\_mirror”。

接下来继续在修改器面板添加一个 Array (阵列) 修改器，选择类型为 Fixed Count 定量复制，在下面的 Count (数量) 中输入 5，并且激活 Merge (合并) 和 Add Offset Object (添加偏移



图 13-88 添加 Empty 作为变形控制器

量控制物体) 两个选项, 在下面的偏移量控制物体输入框中, 输入刚才新建的 Empty 物体名“tir\_mirror”, 设置如图 13-89 左图所示。接着回到视图窗口中, 使用 RMB 选择“tir\_mirror”, 单击组合键 S+Y 做 Y 轴上的旋转, 当旋转 72 度后, 就会发现整个轮毂将按照 Empty 旋转产生的角度偏移量, 自动地被复制出来了, 效果如图 13-89 右图所示。

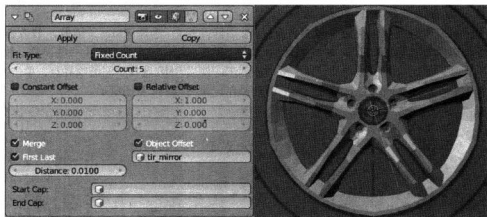


图 13-89 添加阵列修改器

最后在接口处制作一个螺母，由于镜像修改器的作用，只需要制作一边即可，如图13-90左图所示。单击 Tab 回到物体模式，将光标放置在轮毂的圆心位置，然后单击 Shift + A 添加一个 Circle（圆圈）物体，使用 E + S 进行多次挤压后即可得到轮轴物体，如图13-90右图所示。

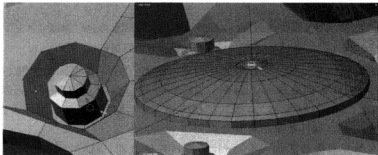


图 13-90 制作螺母和剩余部件

### 13.5.3 制作轮胎

现在开始制作轮胎，在侧视图中新添加一个 Plane 物体，然后使用环切和细分工具挤出胎面的凹凸面，结构如图 13-91 上图所示。这一步很简单，没有特别的技巧，但是一定要注意不要使用太多三角面，因为这会影响最后的细分效果。最后为其添加一个 Mirror（镜像）修改器，设置 Axis（镜像轴）为 X 轴，并开启 Clipping 功能，效果如图 13-91 下图所示。

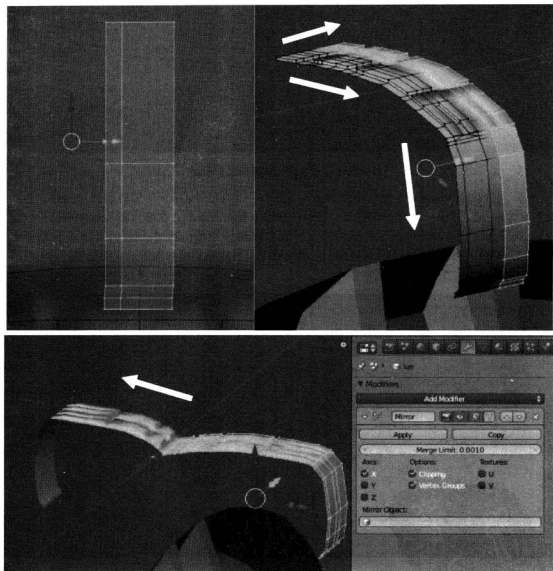


图 13-91 使用镜像修改器制作胎面

接着再添加一个 Array（阵列）修改器，选择类型为 Fixed Count（定量）复制，输入复制次数为 48，同时激活 Merge（合并）和 First Last（首尾连接）选项，设置在 Y 轴上的复制偏移量为 0.52 个单位，这个数字可根据模型的实际情况来调整，最后效果如图 13-92 所示，车胎表面被自动复制出来了。

切换至侧视图视角，首先按照定位轮毂中心的方法，使用 Snap（吸附）工具，将 Cursor

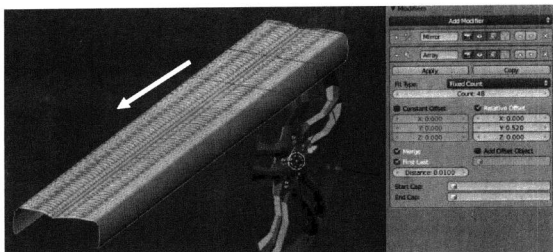


图 13-92 添加阵列修改器延长胎面

(光标)定位在轮子的中心。然后单击 Shift + A 添加一个 Bezier Circle (贝济埃圆) 物体, 将其命名为 “tie\_c”, 如图 13-93 左图所示。接着选择刚才制作的胎面, 在修改器面板中添加一个新的 Curve (曲线) 修改器, 在其面板 Object 控制对象属性中输入刚才命名的贝济埃圆物体名 “tie\_c”, 如图 13-93 右图所示。可以看到, 胎面按照贝济埃圆的形状执行了弯曲变形, 通过缩放圆的大小即可缩放整个轮胎的尺寸。

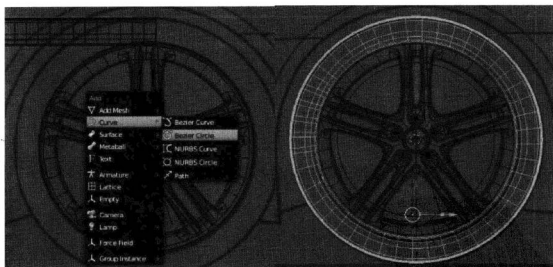


图 13-93 通过曲线修改器对胎面做变形操作

最后, 退出至物体模式下, 在光标位置处, 添加一个新的 Empty (空) 物体。按住 Shift 多选刚才做好的车轮和车胎等物体, 再单选新建的 Empty 物体, 单击 Ctrl + P, 将所有的物体都绑定为 Empty 的子物体, 如图 13-94 左图所示。至此, 一个完整的、可控制的车轮就算做好了。接着切换至侧视图中, 使用 Shift + RMB 框选整个车轮, 按住 Alt + D + G + Y 做属性镜像复制, 并移动至后轮的位置, 如图 13-94 右图所示。

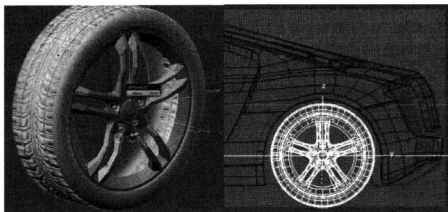


图 13-94 复制出后轮

## 13.6 硬边处理

### 13.6.1 使用切割细分来制作硬边

回到车身模型，接下来需要对表面做进一步的细分处理。

在车体建模时，所有的部件都被建立在一个对象物体中，为了方便编辑和管理，需要将它们按照零件属性互相分离开。这样做的另外一个作用，是可以便于为各部件设置不同的细分等级，例如对于车身可以设置细分等级为3级，而车窗则只需要设置为2级。

选择车体的所有面，单击快捷键P调出分离选项，选择分离 Selection（已选择）的面，如图13-95上图所示。同时在物体的修改器面板中添加一个 Subdivision Surface（细分）修改器，在细分等级中设置为2级细分，如图13-95下图所示。

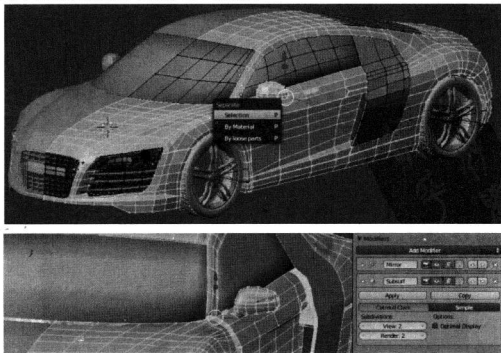


图 13-95 添加细分修改器

但是所有的转角边在细分修改器的作用下，全部变成了圆弧边，如图 13-96 左图所示，这并不是所希望的效果。由于建模时在车身的边缘位置都应用了 Loop 结构，所以现在可以直接在 Loop 结构上使用 Ctrl + R 环切工具，通过添加切割线，并将其放置在 Loop 结构上靠近物体硬边外边缘处的位置，来辅助细分修改器计算出较平滑的硬边效果，如图 13-96 右图所示。

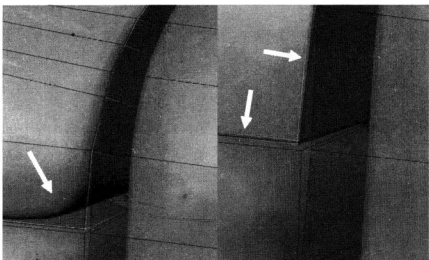


图 13-96 通过切割方式制作硬边

使用同样的方法，在车身上所有的硬边部位都进行环切细分处理操作，如图 13-97 上图所示。很快车体的所有边缘结构就都变得十分平滑细腻了，如图 13-97 下图所示。

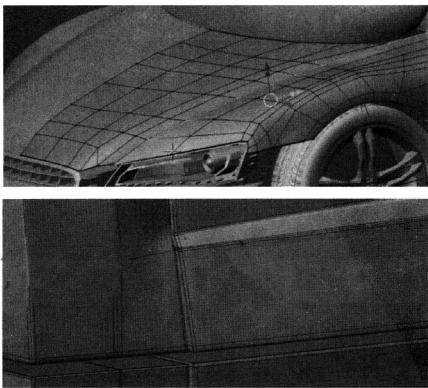


图 13-97 硬边处理后的效果

如图 13-98 所示,为使用 Ctrl + R 添加切割线制作硬边线的位置。有些边还可以尝试添加多条切割线来完成平滑细分。经过这样的处理,模型也将变得更精致了。



图 13-98 硬边处理切割线的布局位置

### 13.6.2 折痕工具制作硬边

当对护栏部件添加细分修改器时会发现,即使是设置等级为 1 的细分修改器,护栏边缘也会由于细分圆滑的作用,变得十分难看,如图 13-99 左图所示。对于这么多细小的多边形,如果使用 Ctrl + R 添加切割线的方法来制作硬边,将会是件工作量十分庞大的事情,同时也会增加很多不必要的面。这时可以使用另外一种处理硬边的办法,首先全选网格物体的所有面,单击 Ctrl + E 调出边菜单,选择 Edge Crease (折痕) 工具,向外拖动鼠标。这时就会发现,护栏物体弱化了修改器对其边缘位置的圆滑影响,表面变得棱角分明,如图 13-99 右图所示。

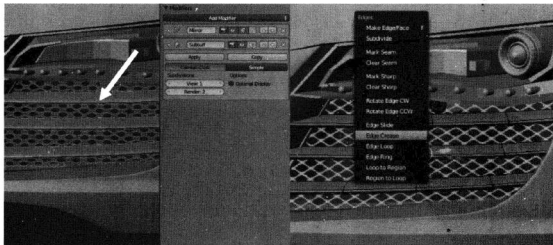


图 13-99 使用 Edge Crease (折痕) 工具来处理简单的硬边

同理,为轮毂物体添加细分修改器,设置细分等级为 2。如图 13-100 所示,是使用 Ctrl + R 处理硬边后的最终效果。

最后添加一个奥迪的四环标志和车灯玻璃罩等附件物体,并且在车轮中添加刹车盘和轮轴等物体,如图 13-101 上图所示。图 13-101 下图就是最后的模型效果和线框图。

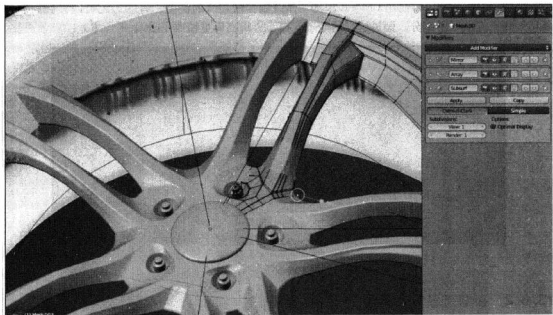


图 13-100 经细分修改器平滑后的轮毂

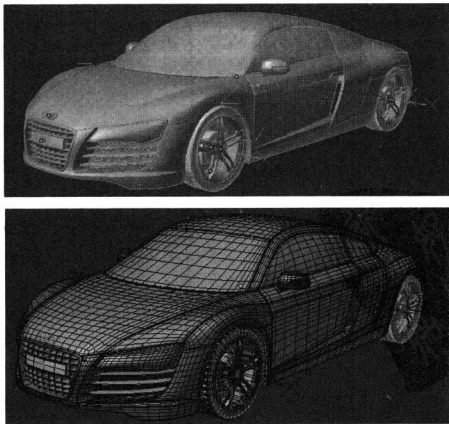


图 13-101 模型效果与线框图



## 13.7 车轮机械约束设计

为了能在动画中模拟车轮转动的机械运动,可以为车轮添加一些约束控制,使车轮在车身的运动过程中可以顺势转动。首先需要对物体进行命名处理,如图 13-102 所示,使用 RMB 选择车身,然后单击 N 在属性窗口中将其命名为 body。

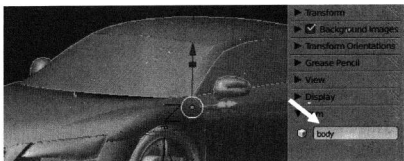


图 13-102 对车体命名

在建模时,曾为车轮添加了一个 Empty (空) 物体,并将车轮的所有零件都绑定为它的子物体,那么在 3D 的机械约束仿真中,就可以通过操纵 Empty (空) 物体,来完成对整个车轮物体的控制。

首先在 Empty (空) 物体的 Object (对象) 属性面板中,找到 Relations (关系) 菜单中 Parent (父系) 子选项,并在下面的输入栏中输入它的父系物体车身的名称“body”,将 Empty 与车身建立起父子关系,如图 13-103 所示。当然,也可以使用快捷键 Ctrl + P 来快速地完成关系绑定操作。当两者产生了父子绑定关系,所有的子物体也就是车轮的全部零件,就会继承父物体车身的运动坐标数据。当车身的坐标发生变化时,它们也会同步产生相对应的坐标偏移变化,场景中的效果就是车身将带动车轮做整体的同步运动。

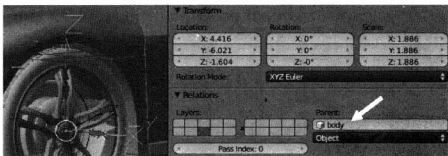


图 13-103 将车轮和车体建立父子关系

接下来,来制作车轮的转动约束控制。选择 Empty (空) 物体,进入 Object Constraints (物体约束) 面板,单击下面的 Add Constraint (添加约束) 按钮,新增一个 Transform (形变) 约束。在面板的 Target (目标物体) 选项中,设置控制形变的目标物体为“body”,目标的 Source (数据源) 为 Loc (位移) 数据,如图 13-104 所示。这样 Empty 就可以直接获取车身坐标的位置运动数据了。

接下来需要根据车身位移坐标的数据变化,来控制 Empty (空) 物体产生转动运动。首先确定车的运动方向,调节 Y 轴上物体运动与约束数据的比例关系,也就是源物体与被控制物体的传动比,设置 Min (最小值) 为 -100, Max 最大值为 100。在 Destination 被影响数据源选项中选择控制方式为 Rot

(转动) 模式, 这样约束器就可以根据源数据的变化来控制当前物体的转动运动了。接下来将 X 轴的运动属性约束在 Y 方向上, 调节约束控制的运动影响范围, 在下面的选项中输入 Min 为 -2500, Max 为 2500。由于需要将全局的数据变换转换至物体的本地运动处理数据, 所以在 Space 约束空间选项中, 选择根据 World Space (世界) 坐标系来约束 Local (本地) 坐标系, 如图 13-104 所示。

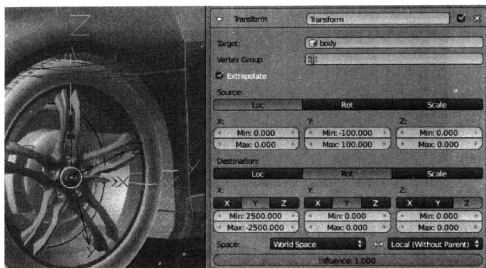


图 13-104 为车轮制作控制约束器

最后, 选取车身物体, 单击快捷键 G+Y 在 Y 轴上移动车身做水平运动, 可以看到车轮将跟着车身向前做滚动运动, 并且可以随着车身的移动速度来调节自身的转动速率, 如图 13-105 所示。同理完成剩下的车轮约束绑定, 模型就可以被应用于动画制作了。

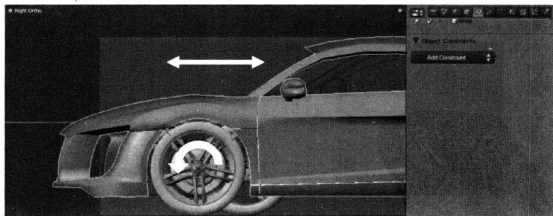


图 13-105 测试转动效果

使用 Blender 提供的各种约束器, 可以制作出其他类似的机械控制效果。感兴趣的读者可以尝试为车身的其他部件添加更多的机械绑定, 例如车门的开启结构、引擎盖的结构、车轮的方向旋转、车轮的避震抖动效果、方向盘与车轮的联动转动等。

## 13.8 材质设计

完成了建模工作, 就可以开始为部件设计材质了。由于在建模时已经将不同的部件分离成

了独立的物体，所以接下来就可以单独选择每个部件，为它们分别赋予材质着色器了。

首先选择车身体，在材质面板中，单击 Add 添加按钮新增一个默认的基础材质。在 Diffuse（漫反射）中选择一个喜欢的烤漆色彩，这里使用 R: 1/G: 0/B: 0 的设置，赋予了纯红色。Specular（高光）色彩选择 R: 0/G: 0/B: 0 的纯白设置，Shade（着色器）类型选择 Cook-Torr 即可，并将 Intensity（强度值）调节为 0.2，Hardness（硬度）设置为 32。在 Shading（着色器）选项中，将 Ambient（阴影色）设置为 1，其他保持默认即可。激活 Mirror（反射）模式，将 Reflectivity（反射强度值）设定为 0.5 左右，色彩保留默认的白色，同时将 Fresnel（菲涅尔）效果设置为 2.3，Blend（混合）属性设置 1.2，具体参数如图 13-106 所示。

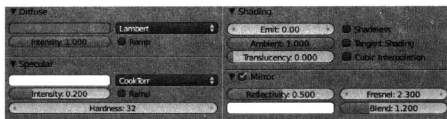


图 13-106 车体材质

接下来制作轮胎和轮毂着色器，因为轮胎是橡胶材质，所以不会有太强的反射和高光效果。在 Diffuse（漫反射）中设置为 R: 1/G: 1/B: 1 纯黑色，将 Specular（高光）设置强度为 0.1，根据橡胶硬度值较低的特性，可以把 Hardness（硬度）设置为 10，具体参数如图 13-107 左图所示。

轮毂为具有一定反射效果的金属材质，所以我们设置 Diffuse（漫反射）为 R0.1/G: 0/B: 0 浅灰色，强度降至 0.8。在 Specular（高光）部分使用 R: 0/G: 0/B: 0 纯白色，Intensity（强度值）调节为 0.5，硬度可设置为 60。打开 Mirror（反射）属性，将 Reflectivity（反射值）增大至 0.9，这里也可以考虑添加一些磨砂效果，在 Gloss（光滑度）面板下将 Amount（数量）降低至 0.9，具体参数如图 13-107 右图所示。

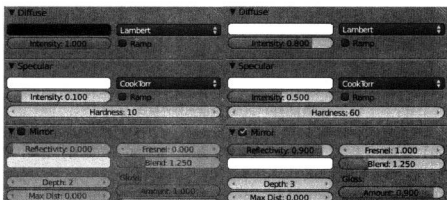


图 13-107 轮胎和轮毂材质

接着选择车窗玻璃物体，在赋予默认材质后做稍许修改。把 Diffuse（漫反射）调节为纯黑色，将 Intensity（强度值）调节至 1。Specular（高光）色彩设置为白色，Intensity（强度值）设置为 1.0，Hardness（硬度值）可调节至 128。打开 Mirror（反射）面板，将 Reflectivity（反射强度值）设置为 0.2，Fresnel（菲涅尔）效果设置为 2.0，Blend（混合）效果设置为 1.25，具体参数如图 13-108 所示。

大灯透明罩的材质与车窗玻璃稍有不同，将 Diffuse（漫反射）设置 Intensity（强度值）为 1.0，Specular（高光）可调至 128。启用 Transparency（透明度）选项，设置为 Z Transparency 即 Z 轴向透



图 13-108 车窗玻璃材质

明模式，将 Alpha（透明度）降至 0.3，Fresnel（菲涅尔）效果提高到 2.0。在 Mirror（反射）选项中调节 Reflectivity（反射值）为 1.0，Fresnel（菲涅尔）效果为 1.0，具体参数如图 13-109 所示。

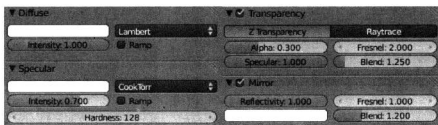


图 13-109 车灯材质

接着选择车上的一些金属部件，例如四环标志等，这些物体都需要较强的反光效果，那么可以为其设置一个 Intensity（强度值）各为 0.5 左右的 Diffuse 和 Specular 材质，同时在 Mirror（反射）属性中调节 Reflectivity 为 1.0，Depth（深度）为 3 级，这样的金属反射会有层次感，具体参数如图 13-110 所示。



图 13-110 高反射的金属部件材质

而深色的通风口网格部件则不需要太强的反射效果，可以设置其 Diffuse 和 Specular 的 Intensity（强度值）均为 0.5 左右，Hardness（硬度值）为 36。激活 Mirror（反射）选项，将 Reflectivity 设置为 0.4，反射色彩为白色即可，具体参数如图 13-111 所示。



图 13-111 深色金属材质

对于车灯的灯管材质，可以添加一个 Intensity（强度值）为 0.5 的金色 Diffuse 材质，设置 Specular（高光）强度值为 1，并在 Shading（着色器）中将 Emit（自发光）选项设置为 0.3，通过自发光来模拟灯泡的照明效果，具体参数如图 13-112 所示。

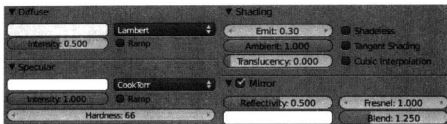


图 13-112 车灯等细节材质

## 13.9 灯光设计

在室内商业产品拍摄中，为了表现一些特殊的视觉效果和氛围，常常需要人为地在物体表面打出特定的光影效果。这些高光和阴影效果的角度和位置，也在一定程度上决定了作品的美感和品质。首先看看一个实际工作室中拍摄汽车外形的布灯设计，如图 13-113 所示。在这个范例中，可以发现，部分车体在灯光的作用下呈现出清晰而富有质感的高光区域，这些效果不是由闪光灯一次性照射出来的，而是使用了经过二次反射的大面积积和面光源反射上去的效果。这样的面光源照明方法是工作室中一种常用的布灯选择，它的光线效果十分柔和和自然，并且可以很容易地控制高光照明边缘范围。



图 13-113 真实环境中的拍摄布光

在 3D 环境中，没有必要去模拟这种二次反射的灯光模式，只需要按照类似的原理，在物体附近放置较大照明面积的面光源即可，如图 13-114 所示。使用一个 Area（面）光源来替代这种照明模式，并根据场景的大小来调节灯光的尺寸与强度。

针对这个场景的尺寸，设置面光源的照明 Energy（光强）为 1.2，Distance（光照距离）设置为 16.56，激活 Specular（高光）和 Diffuse（漫散射）光影响通道，让光源可以产生漫反射和高光效果。在 Shadow（阴影）中激活 Ray Shadow（光线追踪阴影），将这盏灯产生的阴影色彩设置为黑色。采样模式使用 Adaptive QMC（自适应采样），设置 Samples（采样数）为 5，Threshold（阈值）保持为默认值即可，具体参数如图 13-115 所示。



图 13-114 使用面光源作为主光源

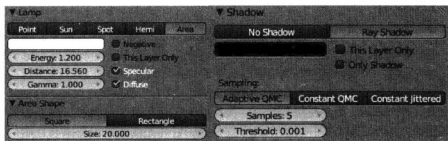


图 13-115 主光源的灯光属性

在车大灯处添加一个 Point Lamp (点光源), 并将 Energy (光强) 设置为 3, 把 Distance (照明距离) 调节至 1.54, 使其光线影响范围仅限于车大灯位置, 具体参数如图 13-116 所示。



图 13-116 添加一个 Point 灯制作大灯效果

## 13.10 环境设置

接着来制作场景, 首先单击 Shift + A 添加一个 Circle (圆) 物体, 使用快捷键 E + S + Z 将其挤压成一个较大的扁圆物体, 如图 13-117 所示。接着为这个地面添加一个默认材质, 将 Diffuse (漫反射) 的 Intensity (强度值) 降至 0.4, 因为不需要地面产生任何高光效果, 所以将 Specular 的 Intensity (强度值) 降至 0 即可。

在面光源的上方再添加一个 Plane 物体, 并使用快捷键 S + Y 做简单的拉伸操作, 然后赋予一个 Emit (自发光) 能量为 2.0 的发光体材质, 将其作为一个反光板物体, 用于为车身提供清晰的高光边缘细节, 如图 13-118 所示。反光板的长宽尺寸可以根据需要的高光效果来决定, 还可以使用多个反光板, 分别摆放在不同的位置和角度上, 用来添加更丰富的细节。

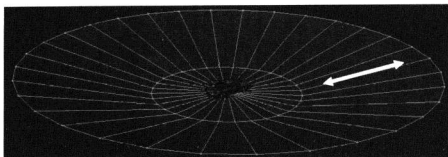


图 13-117 添加一个扁圆作为地面

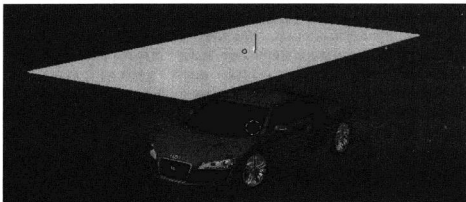


图 13-118 放置反光板

最后在 World（世界）属性面板中，激活 Ambient Occlusion 功能来为场景添加 AO 细节，选择计算模式为 Raytrace（光线追踪），Sampling（采样）模式保持默认的 Constant QMC，其速度快而且质量也不错。Sample（采样数）可设置为 5，Influence（影响值）可修改为 Multiply（正片叠底），这样就可以在画面中均匀添加 AO 追踪计算的明暗对比度效果了，具体参数设置如图 13-119 所示。

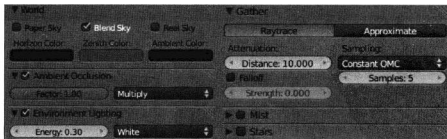


图 13-119 设置 AO 等环境属性

### 13.11 渲染与后期

最后，在场景中添加一个摄像机，并在 Scene（场景）面板中修改 Dimensions（规格）选项下的画面尺寸属性，其中 X 和 Y 分别设置渲染输出的长和宽。单击小键盘的 Num0 进入摄像机视角，使用 Shift + F 飞行模式调节镜头角度，如图 13-120 所示。



图 13-120 摄像机视角

打开渲染面板中的 Anti-Aliasing (抗锯齿) 选项, 激活 Full Sample (全屏采样) 模式, 单击 F12 或者单击 Scene 面板中的 Render Image (渲染图像) 按钮就可以开始渲染了。

待渲染计算完成后, 可以对渲染结果稍作一些后期处理, 激活 Post Processing (后期) 选项中的 Compositing (合成) 功能, 然后进入 Node (结点) 编辑器, 选择状态栏上的 Use Node (开启结点) 系统, 并把编辑模式修改为渲染输出, 刚才的渲染输出结果就会自动导入至编辑器窗口了。

添加一些常用的后期处理结点, 例如借助 Lens Distortion 结点添加镜头的焦距失真效果, 使用 Color Balance 结点做色彩平衡调整, 运用 RGB Curves 结点对对比度做调节, 最后连接至输出结点即可, 如图 13-121 所示。

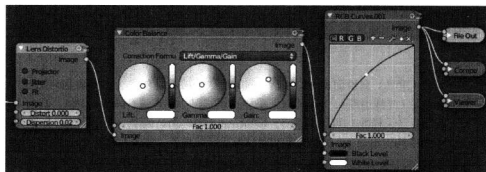


图 13-121 对渲染结果做后期调整

跑车奥迪 R8 的全部制作流程到此就结束, 最后的成品效果图如图 13-122 所示。



图 13-122 奥迪 R8 成品图



## 13.12 本章小结

下面小结一下本章学习中提到的关键点：

- ❑ 建模思路一定要清晰，切忌盲目地直接动手开始建模，适当做一些布线绘制准备工作，可以有效提高工作效率和成品质量。
- ❑ 布线时注意线条的走向应针对不同模型的情况而定，一般来说，线条应贴合物体表面的纹理走向。
- ❑ 建模中应该尽量使用四边形，避免出现五星点或六星点结构，这些点会在表面平滑后带来不小的麻烦。如果一定要使用，请试图尽量把它们放置在不显眼的区域。
- ❑ 理解并熟练使用 Loop 结构。
- ❑ 尽量减小相邻的面之间的面积差距，这样细分后才能保证表面的密度均匀。
- ❑ 选择合理的方法处理物体在平滑后的硬边问题。
- ❑ 使用镜像复制和阵列复制时，应注意物体的原心位置，因为复制修改器将以物体的原心位置为复制的镜像轴原点。
- ❑ 结合父系工具，适当使用 Empty（空）物体来替代约束绑定中的被控对象。
- ❑ 在室内灯光设计时，一定要合理地处理表面的反光效果。可以使用白色反光板来提升反光效果，也可以尝试使用黑板来修正光照的效果。

## 第 14 章

# 角色模型制作

在 3D 应用中,制作人物与角色类模型的难度比制作机械类模型的难度更大,因为制作者不仅需要掌握一定的建模技巧和布线基础理论,还需要熟悉人体解剖学等造型相关知识,才能制作出生动而令人信服的作品。本章以一个人头模型为实例,简单地介绍角色类模型的相关制作流程,身体的其余部位建模将留给读者自己去深入研究。本章重点讲解生物体的建模思路,物体的贴图制作和细节雕刻流程,以及最后的结点合成应用。

## 14.1 造型与布线基础

### 14.1.1 面部造型基础

无论是制作具有真实比例的人物还是虚构的外星怪兽,都需要了解一些头部的比例关系,一般常使用三庭五眼作为标准的比例尺。如图 14-1 所示,为一个标准的人头比例图。

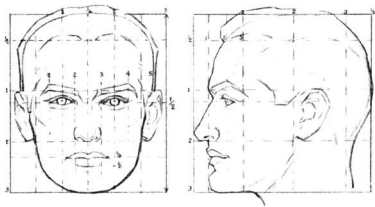


图 14-1 三庭五眼图

三庭指的是人脸的高度刚好等于三个额头的高度,其中头高的一半刚好位于眼睛的中线位置,鼻子的底部刚好位于眉毛到下巴距离的一半处,从鼻底至下巴的一半距离则是下嘴唇的位置,而嘴缝则在鼻底至下巴距离的上三分之一处。从眉毛向上一个鼻子的高度是发际线的位置,耳朵的高度刚好从眉毛到鼻底。

五眼是指脸的宽度刚好为五个眼睛的宽度距离和,嘴巴略比鼻翼宽,整个脸颊大约有三个嘴宽。从侧面看,眼角到嘴角的长度和眼角到耳朵的距离相等。

五官包括眼、鼻、口、耳、眉 5 个主要器官,每一个器官的细节都十分丰富。其中眼部外形

包括眼睑、眼眶和眼球3个部分，鼻子则分为鼻梁、鼻翼、鼻尖、鼻孔、鼻根、鼻背和鼻底7个部分，耳朵分为内耳、外耳和耳廓3个部分，嘴部则主要由唇和齿2个部分组成。

所以在建模时不仅要正确表达五官所在头部的比例和位置，同时还需要强调对每一个细节的深入刻画，才能合理地完成一个生物体结构的制作。

### 14.1.2 布线基础

在对生物体做3D建模时，还需要掌握一些基本的布线法则。

CG也可以被看做是对真实世界中物体的模拟，而其中生物体是自然界中最复杂的物体之一，它的每一个细微的动作和表情都需要数块皮肤下的骨骼和肌肉支撑。如图14-2左图所示，为真实人体面部用于控制表情的主要肌肉分布图。可以发现，面部的肌肉是十分丰富而复杂的，如果要在CG中靠模拟这些肌肉的运动来制作表情十分不现实。因此在为生物体模型做表面布线设计时，只需要按照肌肉和骨骼的生长方向和纹理，制作出近似的线条结构，用来大致使模型调出生动而自然的表情和动作。如图14-2右图所示，为Angela Guenette制作的《Isabella》和角色的脸部布线图。

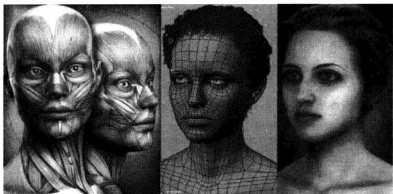


图 14-2 面部肌肉决定了表面布线

但是针对这样的布线，如果表面的面数数量不足，也无法模拟准确的表情。针对于不同领域的应用，角色的线面数量又有不同的需求，例如制作游戏中的人物，为了保证实时运算的性能，物体的网格面数应尽可能地精简。而一般的游戏过场级别CG动画，它在表现力度上有更高的要求，所以单角色的面数要求会达到5万左右。电影工业是对角色细节要求最高的行业，每一个细节被放大到大屏幕上后都要经得起无数观众的检验，这时必须使用几十甚至上百万的网格面才能保证画面不会露出破绽。

生物角色不同于机械，它对表面的布线密度和线条结构也有着不同的要求。对于生物体的表面布线，越密集的布线网络越可以表现较多的细节，同时在伸展变形中也更容易控制拉伸动作。一般来说，运动幅度较大的区域线条应密集一点，例如较活跃的眼部和嘴部肌肉位置。运动幅度较小的地方可使用较稀疏的布线，例如头骨和脑勺位置。

对于生物建模中的线条结构，四边形和四星点是模型中使用最多的基础布局结构单元，它们的大小均匀，排列有序，为后面的细分、蒙皮和添加骨骼绑定等工作提供了很大的便利。但是由于生物体的复杂性，建模中难免会出现五星点和三星点，如何处理这些位置的关系是生物建模中一个重要的问题。

前面曾说过，三星点和五星点在细分后会出现不流畅的平滑表现。其次针对角色模型，非四边形的点和面在表情或者肌肉变形时，相邻的区域会变得难以控制。例如面的运动会在五星点处被迫停止，一个微笑的表情就会因为某一个不规范的点而变得十分僵硬。所以如果在无法避免的情况下，也应该尽可能地将非四星点（也就是三星点或五星点）藏在运动幅度最小的地方，或者视觉范围之外的区域。不过针对游戏等低模要求的角色，就没必要考虑非四星点和平滑后的效果了。

如图 14-3 左图所示, 是一个比较经典的面部布线图, 它遵循了基本的脸部布线原则, 使用 Loop 结构完成面部的肌肉轮廓。如图 14-3 右图所示, 则是使用这种类似布线结构的优秀的作品范例。

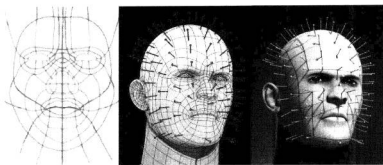


图 14-3 Milan Kalis; Pinhead

## 14.2 人物建模

### 14.2.1 建立参考图

按照第 13 章中介绍的方法为场景建立参考图, 首先按小键盘的 Num1 进入正视图视角, 按 N 激活 Background (背景图) 选项, 单击 Add (添加) 图片按钮, 选择在 Photoshop 中切割好的参考图片, 将 Axis (约束轴) 设置为 Front (正面), 如图 14-4 所示。同理添加一个侧视图, 并将 Axis (约束轴) 设置为 Right (侧面)。

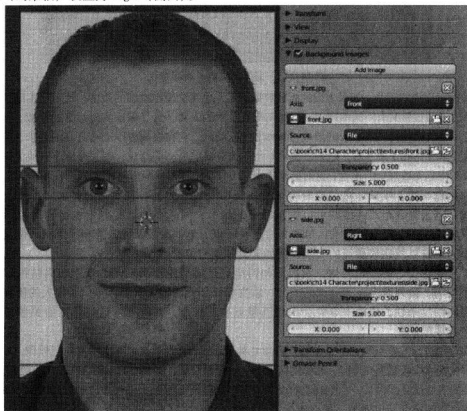


图 14-4 建立参考图

接着新建一个绘制图层，单击 D + LMB 开始绘制布线草图。根据上一节中讲到的布线基本规则，把眼部的线条向四周做发散性分布，同时在鼻子、嘴唇和眼眶部位设计 Loop 环形布局结构，如图 14-5 所示。

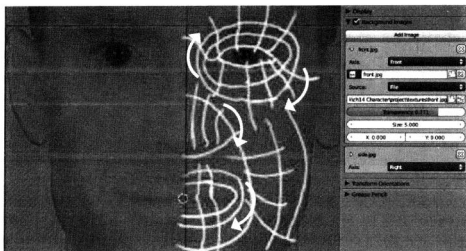


图 14-5 绘制布线草图

## 14.2.2 面部建模

单击 Shift + A 新添加一个 Cube 物体，接下来在 Modifiers（修改器）面板中为其添加一个 Mirror（镜像）修改器，激活 Clipping 选项。单击 X 删除全部的点后，再添加一个 Plane 面，如图 14-6 所示。

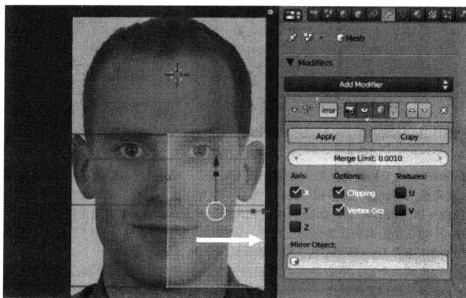


图 14-6 建立 Plane 物体并添加镜像修改器

单击 G 将 Plane 面移动至眼部位置，在面选择模式下，单击 X 删掉中间的面。然后全选所有的点，使用 W 细分工具对其做两次细分操作，配合 G 分别将它们移动到参考图中的位置，效果

如图 14-7 左图所示。接着选择这些点，然后按 E 向外挤压出第一个 Loop 环结构，并参考背景图中的线条布局，调节这些点的位置，最后的效果如图 14-7 右图所示。

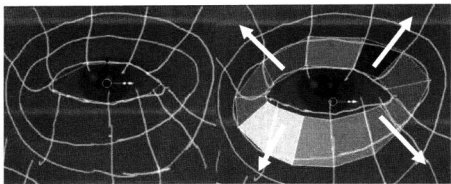


图 14-7 使用 Loop 结构制作眼眶

为了方便制作眼眶的剩余部分，退出脸部的编辑模式，单击 Shift + A 在眼眶位置添加一个新的 UV 球体物体，单击 Tab 进入其编辑模式后，将接口点旋转到 Front 前视图方向，使用 Alt + RMB 来选择一个圆环并向 Y 轴负方向稍作移动，用于制作眼球上的瞳孔，如图 14-8 左图所示。完成后单击 Tab 退出编辑模式，使用 G 将眼睛装配到眼眶的位置，接着调整眼部的网格，使其与眼球的弧度相匹配，效果如图 14-8 右图所示。

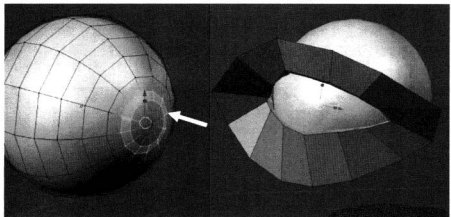


图 14-8 添加眼球

单击 Num3 切换至侧视图视角，选择眼眶周围的点，对其位置进行微调，注意将眼角稍微向后移动，将眼眶的中间部分向前稍微突出，形成与眼球一致的曲面。然后选择最外面的边，单击 E + S 继续向外挤压出 Loop 边，如图 14-9 左图所示。接着单击 Num1 回到正视图视角中，选择靠近鼻梁的一个边，单击 E 向正中做挤压操作，镜像修改器会自动缝合接口部分。然后再围绕眼眶添加更多的线条和 Loop 环切结构，最后的效果如图 14-9 右图所示。

选择下眼眶的两个边，单击 E 向鼻翼方向挤压，如图 14-10 左图所示。当挤压至鼻尖位置时，应在下方鼻孔的位置排列出一个 Loop 结构，同时封闭鼻梁上的点。接着将鼻孔位置的边继续向内挤压，使用快捷键 S 缩小内部的面，最后的效果如图 14-10 右图所示。

单击 Num1 回到正视图视角，挤压并缝合额头部分的面，使用 Alt + RMB 选择最外侧的 Loop 结构边，继续向外稍作挤压，如图 14-11 左图所示。接着使用 MMB 旋转至自由视图视角，缝合前额的面，并对多边形上的各点进行位置微调，例如眼内眦到鼻翼的深度、颧骨和眉骨之间的高

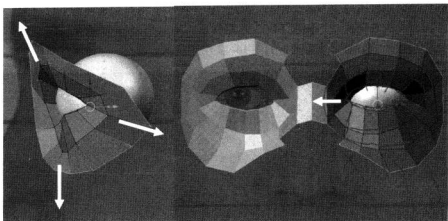


图 14-9 缝合对称面

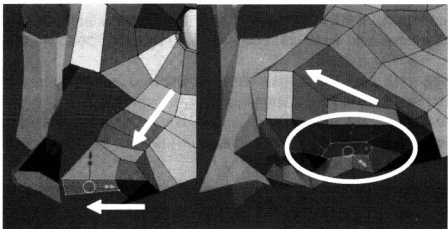


图 14-10 制作鼻子

度，如图 14-11 右图所示。在粗模的制作中，可以采取稍微夸张的手法去处理整体的形态，待后面再慢慢去调整每一个部分的细节。

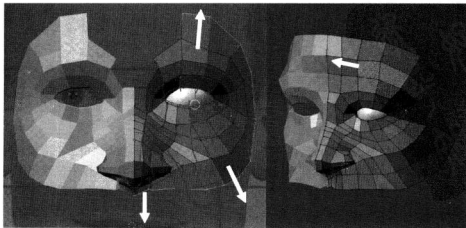


图 14-11 缝合面部

嘴部的口角提肌可由两个 Loop 结构来组成, 第一个 Loop 由鼻梁延伸下巴位置, 这样可利于模型制作出较大的咬合运动, 如图 14-12 左图所示。第二个 Loop 环绕在口轮匝肌部位, 这样能使模型做出细腻的嘴唇咀嚼等运动, 如图 14-12 中图所示。

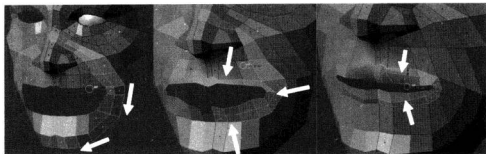


图 14-12 制作嘴部的 Loop 结构

按照这样的思路从鼻侧处向下巴方向挤压, 然后继续选择嘴唇周围的边向内挤压, 使用 G 移动工具调整线条的疏密。利用环切工具 Ctrl + R 在 Loop 结构上添加切割线, 最后制作出嘴唇的大致形态, 如图 14-12 右图所示。

选择脸部模型最外侧的 Loop 边向后做挤压操作, 如图 14-13 左图所示。现在再观察一下成品的模型, 面部已经形成了几个重要的 Loop 结构, 一个在眼眶部位, 一个是鼻侧至下巴的位置, 最后是嘴唇和鼻孔处布置的 Loop 结构, 这些 Loop 结构不仅能方便后面的细分操作, 还有利于模拟表情肌肉为模型制作动画中的变形, 如图 14-13 中图所示。最后, 调整这些边的位置关系, 效果如图 14-13 右图所示。

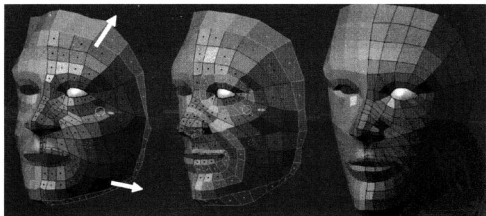


图 14-13 面部的 Loop 结构

从下颚的后部向后脑勺方向做挤压操作, 如图 14-14 左图所示。头顶由于不需要任何细节所以可以放宽面数要求, 简单的挤压和缝合操作即可完成整体的造型。在耳朵位置删掉一些点留出一个圆弧缺口, 这里一会儿需要单独制作出耳朵, 并缝合上去, 最后头部的调节效果如图 14-14 右图所示。



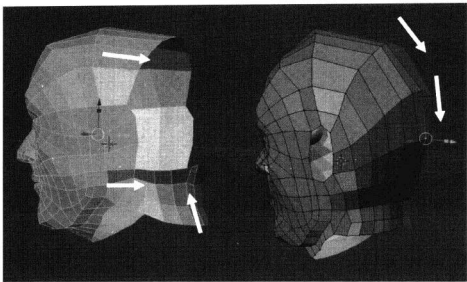


图 14-14 头部的调节效果

### 14.2.3 耳朵建模

耳朵应该是头部建模中最难的一个部分了，它的结构复杂，转折区域较多。但是由于没有较大的运动幅度，所以只需要做出形态即可，无需考虑太多布线与优化的问题。

单击小键盘的 Num3 进入侧视图视角，添加一个 Plane 物体，参照耳廓的结构按 E 挤压一个多边形结构，如图 14-15 左图所示。接下来分别使用 Alt + RMB 选择耳廓内侧和外侧的边，向内外两个方向做挤压操作，如图 14-15 中图所示。最后使用 F 缝合耳朵的内侧部分，布局尽可能地使用 Loop 结构，效果如图 14-15 右图所示。

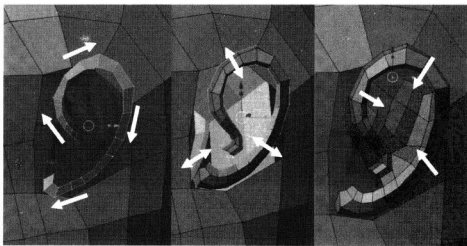


图 14-15 制作耳廓

接下来使用 F 缝合工具，将耳廓的外侧与耳垂部分连接起来，如图 14-16 左图所示。利用 Ctrl + R 做切割细分操作，制作出耳垂部位，并顺势向内挤压出耳洞，如图 14-16 中图所示。为整个头部添加一个细分修改器后，可以预览一下耳朵细分后的效果，如图 14-16 右图所示。

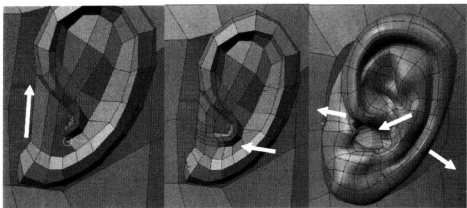


图 14-16 耳朵细分效果

#### 14.2.4 细分优化

接下来需要对五官部分做一些细分处理，根据人物的需求，适当地调整五官的比例和位置。细分的方法建立在 Loop 结构的基础上，使用 Ctrl + R 添加切割线，再使用 G 微调点和线条的位置。如图 14-17 左图所示为眼睛的细分结果，可以在这里切割出两个 Loop 圈。面部中可添加细分结构的部位主要包括眼眶位置、上下眼睑结构、鼻翼内侧和鼻孔四周等，如图 14-17 右图所示。可以在嘴唇处多做几次细分切割线，制作出嘴唇的立体感，同时不要忘记鼻下的人中部位。

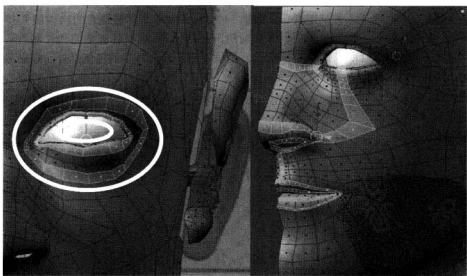


图 14-17 对眼部和嘴部做细分处理

在保证五官各部分面密度均匀的基础上，围绕眼眶和鼻梁部位添加更多的细分线，如图 14-18 左图所示。接着将耳朵与头部缝合起来，由于耳朵部位没有动作要求，所以可以使用部分三星或五星点的缝合接口。如图 14-18 中图所示，为细分修改器下的预览效果，可以看到眼和嘴的线条最密集，而头部和脖子部位的线条较稀疏，这样的表面密度刚好符合我们对布线标准要求。因为这不是一个复制真人的建模，所以背景图仅供参考。对模型再做些自定义的调整，最后的效果如图 14-18 右图所示。

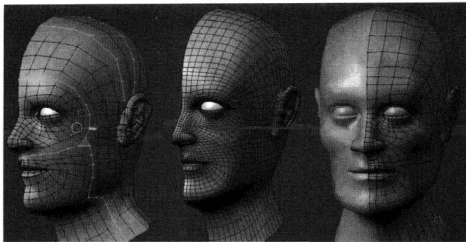


图 14-18 细分调整后的脸部效果

### 14.3 头骨建模

为了讲解如何在 Blender 中使用雕刻功能，在刚才的人物模型基础上，制作一个稍微科幻点的人物：Two Face（双面人）。

双面人特效已经出现在多部好莱坞的电影中，其中最著名的特效效果当属《终结者》中被毁容后的半金属半人类的 T-800 机器人施瓦辛格，如图 14-19 左图所示，以及《蝙蝠侠 黑暗骑士》中被大火烧伤的高谭市检察官哈维邓特，如图 14-19 右图所示。在神奇的电影 CG 特技下，主人公的半边脸呈现出十分令人惊悚的效果：被掏空的下颚，完全暴露的眼球组织，撕裂的皮肤和白森森的头骨。接下来就在刚建好的基础模型上做一些修改，为雕刻双面人做一些准备工作。



图 14-19 Two Face（双面人）

#### 14.3.1 制作面部缺陷效果

首先在 Object Mode（物体模式）中，选择人头模型，单击 Modifier（修改器）中的 Apply 按钮，将镜像修改器的对称修改应用至模型上，如图 14-20 左图所示。接下来更换一个参考图，导入一个头骨解剖参考图，调整它的位置和尺寸，如图 14-20 右图所示。

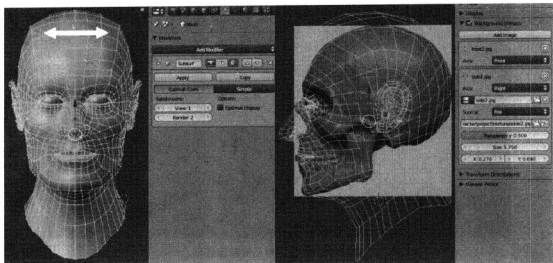


图 14-20 应用镜像修改器并添加新的参考图

由于设定双面人的左脸没有眼睑和眼轮匝肌，所以在正视图视角中，可以直接删掉左眼的大部分结构面，如图 14-21 左图所示。接着选择眼眶的边，根据眼球的位置重新调整表面 Loop 的结构，并按 E 向内挤压至眼球后方，使整个眼球都从眼窝中凸显出来，如图 14-21 右图所示。

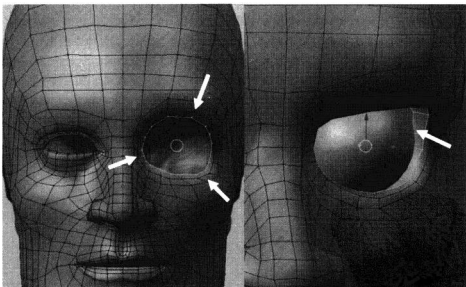


图 14-21 修改眼眶

由于左脸的口轮匝肌和嚼肌也所剩无几，为了精确地切割出这些缺口，首先激活 Grease 工具，勾选 Drawing Setting（绘制选项）中的 Surface（表面）模式，按住 D + LMB 在嘴部位置画出一个预期的切口边，旋转视图后就会发现线条已经吸附在面部表面上了，如图 14-22 左图所示。为了在切口处整理出 Loop 结构，首先使用快捷键 K 做精确切割操作，然后围绕草稿线的周围添加一些必要的点来调整出一个 Loop 结构，并使用 F 将两个相邻的三角面缝合成一个四角面，如图 14-22 右图所示，注意箭头处的处理点。最后删掉中间的点，完成脸部的空洞效果，如图 14-22 下图所示。

为了突出下巴位置露出的下颌骨，在下巴处也挖一个洞，对划线区域的面做两次精确切割，

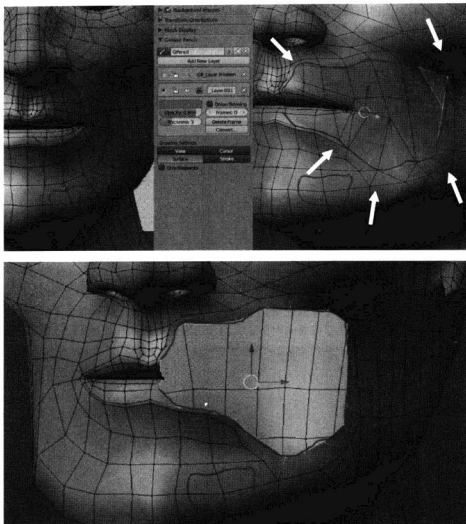


图 14-22 制作腮帮处的空洞效果

如图 14-23 左图所示。然后使用删除和重缝合的方式，将切割出来的点和面整理成一个 Loop 结构，如图 14-23 中图所示。使用类似的方法，在鼻侧处挖出一个洞，用于凸显鼻腔内的结构，如图 14-23 右图所示。

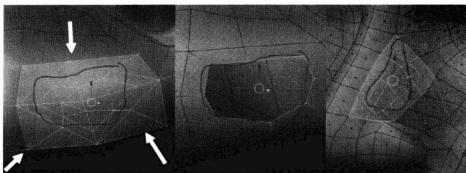


图 14-23 使用精确切割来制作缺口

可以在头颅的侧面再剥离些皮肤用于突出颧骨。使用快捷键 D + LMB 描绘出可吸附的草稿线, 用于辅助操作, 如图 14-24 左图所示。选择这些面后使用 K 做精确切割, 在切割线的两头可以保留三角面, 选择内边再按 G 向颅内移动, 调整结果如图 14-24 右图所示。

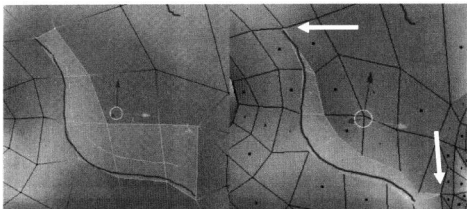


图 14-24 切割出颧骨处的细节

接着, 使用同样的方法, 在头顶处做一些切割和挤压, 制作一点凸起的 Loop 结构, 如图 14-25 左图和中图所示。然后在腮帮处向内挤压, 制作出塌陷的效果, 如图 14-25 右图所示。

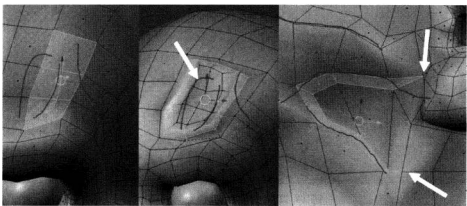


图 14-25 添加头顶和下颚骨的细节

### 14.3.2 制作口腔结构

单击 Num1 切换至正视图视角, 退出头部物体的编辑模式, 添加一个新的 Cube 物体, 然后按 W 选择平滑细分, 并删掉顶上的面, 调整成牙齿形状, 如图 14-26 左图所示。接着使用 Shift + D 将物体做多次复制, 并按 G 将其排列成如图 14-26 右图所示的形状。可根据角色的需要随意发挥牙齿的形状和排列方式, 而无需保持整齐的造型效果。

完成了牙齿的编辑后, 单击 Shift + A 在牙床的附近添加一个新的 Plane 物体, 调整形态后做挤压操作, 准备制作牙龈体, 如图 14-27 左图所示。这里没有太多的技巧, 只需要细致一点就可以挤压出如图 14-27 右图所示的牙龈部分, 以及旁边的下颌骨结构。

从正视图视角上, 再对模型的骨骼和结构做一些戏剧性的调整, 例如调整眉骨的造型曲线, 以便突出一种邪恶的表情, 突出颧骨的高度和嘴角塌陷的效果, 用于强化角色脸部的恐怖感, 最后的效果如图 14-28 左图所示。完成之后使用 Alt + RMB 选择伤口处的所有边, 按 E 向内挤压, 制作出面部暴露肌肉的厚度感, 如图 14-28 右图所示。

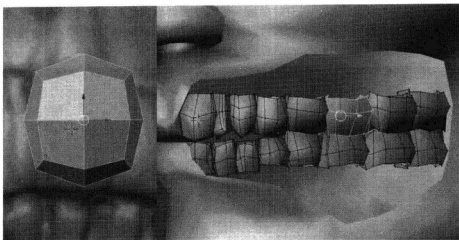


图 14-26 制作牙齿

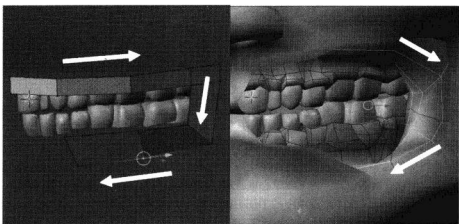


图 14-27 制作牙龈和下颌骨

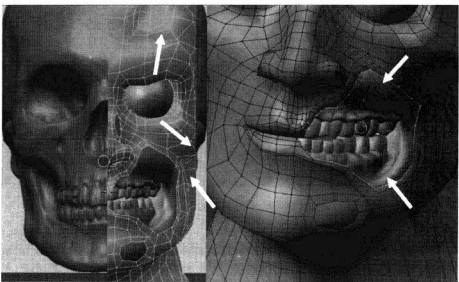


图 14-28 调整造型

打开细分修改器的预览模式，再对外形做一些调整，例如调整肌肉厚度与头骨之间的关系，如图 14-29 左图所示。最后在牙龈多边形的基础上添加一个 Cube 物体，挤压后移动到口腔外侧，制作出残存暴露的嘴唇，如图 14-29 右图所示。

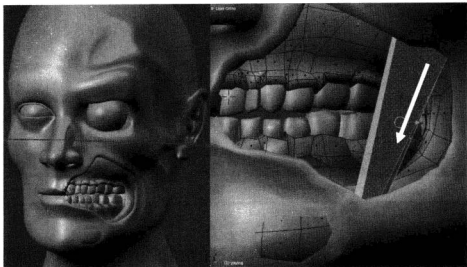


图 14-29 制作嘴唇

### 14.3.3 制作眼球控制器

接下来需要对眼球添加一个方向约束，用于控制目光的注视方向。

首先应用眼球的镜像修改器，然后选择单个眼球物体，按 P 将两个眼球分别独立出来。这时眼球的物体原心仍位于独立前的位置，为了将原心重新定位在球体正中，退出至眼球的物体模式，同时选择两个物体，单击 Object（物体）选项中的 Transform（形变）菜单，如图 14-30 左图所示。在二级目录中选择 Origin to Geometry，将物体原心初始化至眼球的圆心处，最后的效果如图 14-30 右图所示。

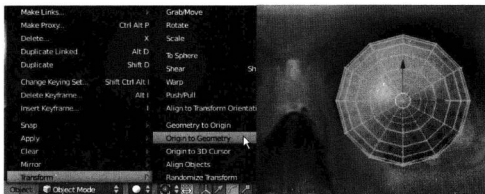


图 14-30 重置眼球的圆心位置

在物体模式下，选择任意一个眼球，单击 Shift + S 调出 Snap 吸附菜单，选择 Cursor to Selected 将 Cursor（光标）移动到眼球中心，如图 14-31 左图所示。接着在光标的位置单击 Shift + A 添加一个 Empty（空）物体，如图 14-31 右图所示。完成对两个眼球上空物体的添加后，将它们分别命名为 left\_eye 左眼球、eye\_left\_cont 左眼球空物体控制器、right\_eye 右眼球和 eye\_right\_cont 右



眼球空物体控制器。

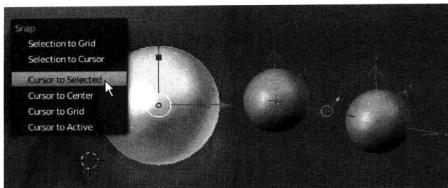


图 14-31 在圆心位置添加 Empty（空）物体

同时选择两个 Empty（空）物体，单击快捷键 G+Y 向 Y 轴的负方向移动一段距离，接着选择眼球物体，在 Constraint 约束面板中为其添加一个 TrackTo 跟随约束。以左眼为例，在约束的 Target（目标）中输入左眼球控制物体的名称 eye\_left\_cont，将约束方式固定为 Y 轴，具体参数设置如图 14-32 所示。

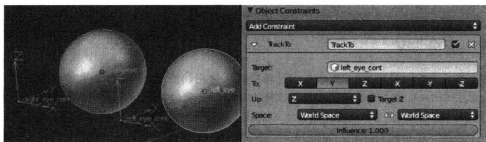


图 14-32 制作眼球的控制约束器

同理，完成对右眼球的约束控制器添加，接着在两个 Empty 控制器正前方处添加一个新的 Empty（空）物体，命名为 eye\_main\_control 目光主控制器。选择两个眼球控制器，再选择目光主控制器，单击 P 为它们建立起父子关系。这样只需要移动 eye\_main\_control 目光主控制器物体，就可以同时控制双眼的注视方向了，如图 14-33 所示。

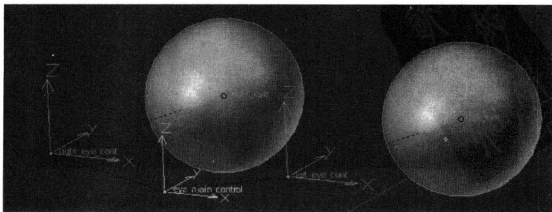


图 14-33 添加目光的跟随控制

最后的效果如图 14-34 所示。基础建模部分就算完成了。

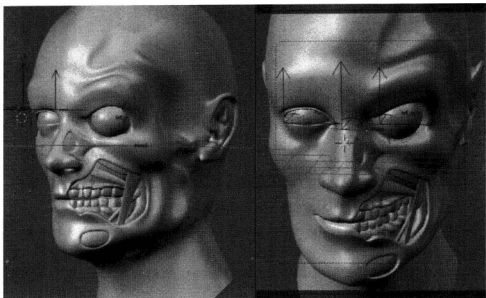


图 14-34 完成的基本模型

## 14.4 拆分 UV

接下来，开始对模型拆分 UV。在边选择模式下使用快捷键 **Alt + RMB** 选择头部的所有开放接缝，单击 **Ctrl + E** 调出边的编辑菜单，选择 **Make Seam** 将边标记为缝合边，如图 14-35 左图所示。再从人头正中处添加一个接缝线，将耳朵从根部切开与头部分离，同时在脖子与头部的位置添加一个接缝，也将两者分开，如图 14-35 中图所示。完成后全选模型，单击 **U** 弹出 **UV Mapping** 拆分映射菜单，选择 **Unwrap** 解包拆分 UV，如图 14-35 右图所示。

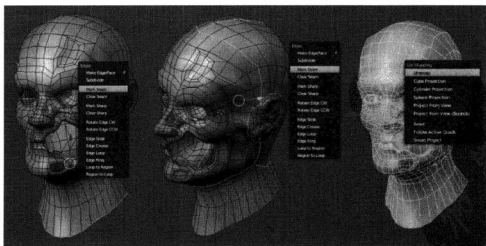


图 14-35 标记缝合边并拆分 UV

在工作区域中切出一个新窗口，选择视图模式为 **UV/Image Editor**，就可以看到自动解算出的 UV 图了。单击 **N** 调出 UV 属性面板，激活 **Stretch** 伸展选项，将 UV 中不合理的拉伸幅度用彩色

标识出来,使用RMB选取UV上的点并单击G稍作微调。其中蓝色代表这块区域的UV无拉伸,颜色越绿甚至变为红色,代表当前位置的UV出现了大幅度的拉伸。拉伸幅度过大的UV将影响最后的贴图效果,所以应该对UV做细微的调整,尽量减少UV面的拉伸效果。

为了辅助调节,可以单击Alt+N为当前UV新建一个贴图,并在弹出窗口中激活UV Test Grid选项设置贴图为网格检查图,并在物体模式中激活Texture(纹理)显示模式,这样就可以在3D窗口中实时预览UV的调节效果了,如图14-36所示。一个好的UV拆分结果,应当使测试图中的网格在物体表面上的显示呈大小均匀、排列整齐的效果。

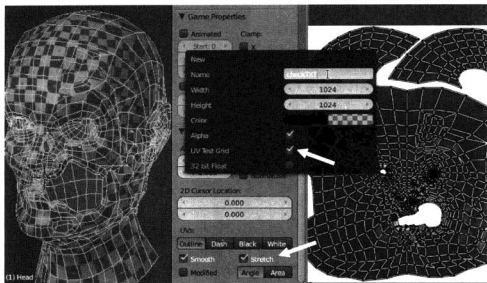


图 14-36 添加网格图来检查UV的拉伸效果

同样对剩余的部位(例如牙龈)做简单的UV拆分。首先在边选择模式下,单击Shift+RMB选择任意一个未缝合的边,然后按Shift+G调出相似选择菜单,单击Amount of Vertices in Face可快速地选择所有相似外缝合边,如图14-37左图所示。最后按Ctrl+E将它们标记为缝合边,并按U做UV拆分,拆分的效果如图14-37右图所示。

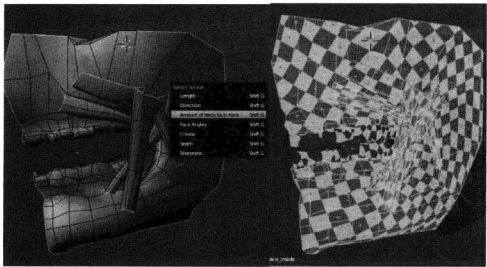


图 14-37 对牙龈部分做UV拆分

## 14.5 模型雕刻

在对模型做雕刻操作前,首先来对物体做一些细分处理。选择头部模型,在修改器面板中添加一个 Multi-Resolution 修改器,单击 Subdivide (细分) 按钮,将 Sculpt 雕刻细分等级提升至 3 级,并同时设置 Preview 视图等级也为 3 级,如图 14-38 所示。

选择头部物体,在操作模式菜单中选择 Sculpt 雕刻模式,鼠标将显示为当前笔刷的尺寸,如图 14-39 左图所示,这时就可以开始正式雕刻了。在刚开始的时候,不需要去描绘太多的细节,只需要把整体的形态刻画出来,并控制骨骼上肌肉的纹理走向即可。首先使用默认的 Add (添加) 画笔,配合快捷键 F 调节笔刷的大小,以及 Shift + F 调节笔刷的强度,在模型上做一些基础的造型雕刻,效果如图 14-39 右图所示。

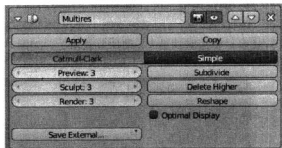


图 14-38 使用 Multi-Resolution 修改器对模型做雕刻前的细分

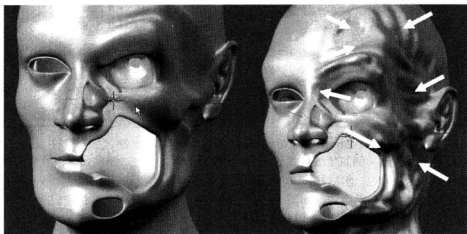


图 14-39 雕刻大体形态

完成了整体的造型雕刻,接着再将细分等级提升至 4 级,然后切换至 Sub 画笔,在模型上做深入处理,制作出面部的凹凸细节,如图 14-40 左图所示。为了表现面部有被撕裂的痕迹,应使用较小笔触的画笔来刻画细节,如图 14-40 右图所示。

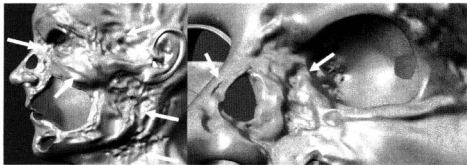


图 14-40 提升细分等级并雕刻局部细节

为了找准物体之间的关系，可以打开眼球和嘴部肌肉等物体的显示，在有物体相互遮挡的情况下做进一步的细节雕刻。对之前错误的布局和细节可使用 Smooth（平滑）笔刷来调整，如图 14-41 左图所示。在这一步就需要确定整体结构的分布，对不恰当的设置要勇于去平滑或者删除，调整后的效果如图 14-41 右图所示。

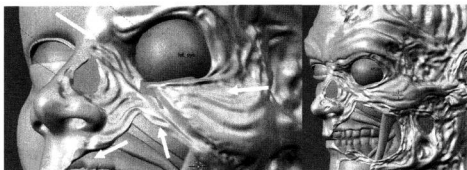


图 14-41 进一步的细节雕刻

完成了左脸的雕刻，使用 MMB 旋转视图，在右脸上做肌肉和表情的雕刻。为了突出人物的性格特征，可以使用夸张的肌肉动作来塑造情绪。当人们在表达愤怒时，面部会出现横眉立目，挤眉弄眼，咬牙切齿等表情，同时面部的肌肉由于紧张会绷紧起来，并产生出更多的表情细节，例如嘴角下撇，鼻孔张大，双眼微闭等。

如图 14-42 左图所示，首先使用大尺寸低强度的 Grab 笔刷，通过推挤操作来调整嘴部和脸部肌肉的动作。这时就能体会到布线对表情调节的影响，如果没有使用 Loop 结构来制作脸部的嚼肌肌肉，嘴角的皮肤叠加动作将无法得以正常表现。接下来再调低笔刷尺寸，并使用较大的雕刻力度来加深肌肉的深度，最后的效果如图 14-42 右图所示。

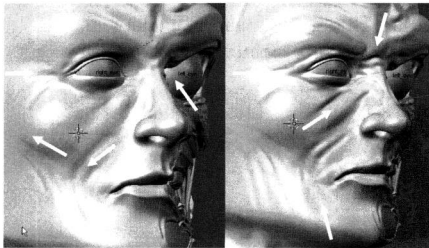


图 14-42 刻画表情

配合 Pitch 笔刷来挤压出嘴唇等部位的边缘，并使用 Sub 笔刷雕刻出嘴唇上的纹理，如图 14-43 所示。

完成了整体造型的雕刻，接下来就开始对细节做更深入的刻画。首先将模型细分等级提升至 5 级，接着为 Sub 笔刷添加一个 Texture（纹理），选择一个预先制作好的皮肤纹理，并设置



图 14-43 制作嘴部细节

Stroke 模式为 Anchored，如图 14-44 左图所示。接着单击 Shift + F 降低笔刷的力度，在模型表面上做拖动操作，绘制出皮肤的纹理细节，效果如图 14-44 右图所示。

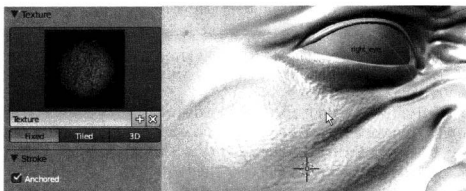


图 14-44 使用纹理映射来雕刻皮肤质感

针对左脸的表皮，可以切换至不同的纹理图片，并在面部拖动出更多的细节，如图 14-45 所示。

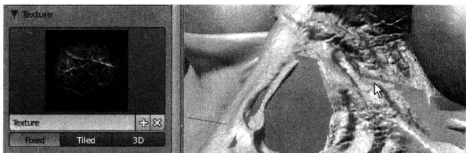


图 14-45 切换笔刷纹理来雕刻出不同细节

细节的刻画应注意疏密有度，并不是每一个位置都需要十分丰富的深度表现。根据设定，人物的左脸属于部分毁坏，因此在雕刻的过程中，应保持整体与细节的关系，不要用过多的小纹理去掩盖掉全部肌肉的形态。如图 14-46 所示为完成后的眼部细节。

通过不断的调整，丰富可见部位的细节，就完成了物体的雕刻工作，最后左脸与右脸的雕刻

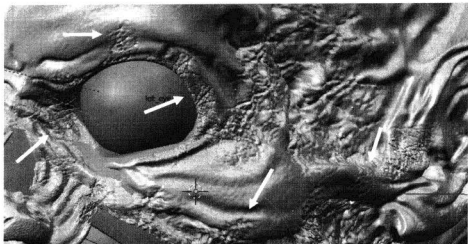


图 14-46 完成后的眼部细节

效果如图 14-47 所示。接下来按照类似的工作流程，完成剩余部分的雕刻工作，例如嘴部的肌肉，这一部分的雕刻留给读者完成，本章中不再详细描述。

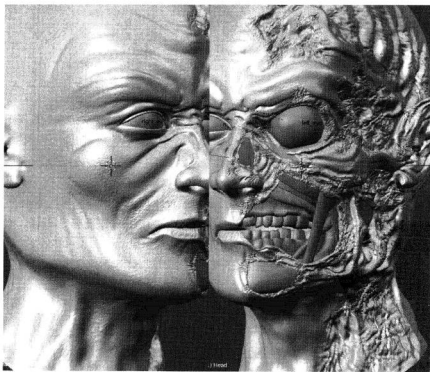


图 14-47 面部模型的雕刻成品效果

## 14.6 烘焙贴图

由于最后完成雕刻的高模多边形数量十分惊人，要直接运用在渲染中将花费大量的计算时

间。为了获得性能与效果的平衡,接下来就需要使用烘焙工具,将高模的凹凸纹理,映射至 2D 贴图中,然后再使用贴图的法线计算模式,在 3D 视角中显示出这些凹凸的细节效果。

首先复制一个头部模型物体,将其雕刻细分降低至 1,然后单击 Tab 进入编辑模式,在 UV 编辑器中,为其新建添加一个空白贴图,设置分辨率为  $1024 \times 1024$ ,如图 14-48 所示。

接下来,使用 RMB 选择细分等级为 5 的高模物体,再按住 Shift,选择带贴图的低模头部模型物体,如图 14-49 所示。

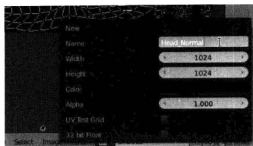


图 14-48 为低模头部物体添加新建贴图

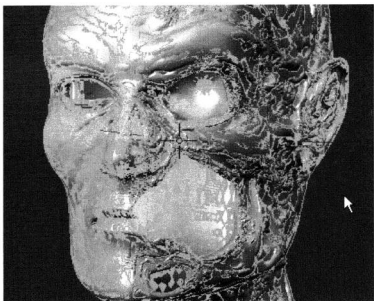


图 14-49 同时选择高模和低模物体

切换至 Scene (场景) 面板的 Bake (烘焙) 菜单。这里提供了多种烘焙渲染模式,首先制作 Normals (法线) 贴图。将 Normal Space (法相空间) 设置为 World (世界) 坐标,激活 Selected to Active 选项,使烘焙仅计算当前选择激活的物体,具体参数设置如图 14-50 所示。



图 14-50 设置 Bake 烘焙选项



接下来单击 Bake（烘焙）按钮就可以开始烘焙贴图了。经过一段时间的运算后，就可以在低模物体的 UV 编辑模式下，查看到烘焙成功的法线贴图了，如图 14-51 左图所示。同理，还需要烘焙 Bump（凹凸）纹理贴图和 AO 环境阴影贴图，最后的效果分别如图 14-51 中图和右图所示。关于这几种贴图的使用，将在后面的材质设置中重点讨论。



图 14-51 烘焙出 Normals（法线）贴图、Bump（凹凸）贴图和 AO 环境阴影贴图

## 14.7 映射绘制

完成了纹理等质感类贴图的制作，现在开始制作色彩贴图，也就是 Diffuse 通道贴图。

绘制色彩贴图的方式有几种，最常用的方法是将物体的 UV 展开成线框图输出，然后导入其他图形编辑软件（例如 Gimp 和 Photoshop）中，在这些软件中直接绘制纹理的色彩。但是这样的绘制方法很容易出现表面纹理的拉伸问题，以及贴图在物体表面的位置显示错误，特别是接缝边处的纹理。

另一种方法则是使用 Blender 提供的 Projection Painting（映射绘制），它将使用投影的方式将视图画面直接导入绘图工具中，从物体的视角上做纹理的绘制，待纹理绘制完成后，再将其映射回物体的表面上。这种方法增强了编辑的实时性，同时也减少了贴图的拉伸错误和位置混乱等问题。

首先，和新建烘焙贴图的流程一样，需要在较低细分等级的头部模型编辑模式下，新建一个 1024 × 1024 尺寸的贴图文件，并设置底色为白色，如图 14-52 所示。完成了贴图添加后，需要重新将物体的细分等级提升至最高级。

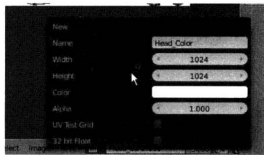


图 14-52 新建绘制贴图

接着进入系统的设置面板，或者单击快捷键 Ctrl + Alt + U，在 File 文件一栏中的 Image Editor（图片编辑器）设置中，输入将使用的平面绘图软件目录地址。例如本例中使用的是 Photoshop，设置如图 14-53 所示。

回到 3D 视图中，将物体的编辑模式切换至 Texture Paint（纹理绘制）模式，单击 T 调出绘制工具面板，在其中激活 Project Paint（映射绘制）选项，接着将映射尺寸修改为和贴图同样大小的 1024 × 1024 像素，如图 14-54 左图所示。最后调整视图中物体的角度和位置，然后单击 Quick Edit（快速编辑）按钮，Blender 将自动启动刚才设置的图片编辑软件 Photoshop，同时将当前 3D 视图场景中物体的画面导入编辑界面中，如图 14-54 右图所示。

接着借助物体的截图画面为参考图，在 Photoshop 中新建一个图层，并开始绘制面部的纹理贴图。这里可以导入一个现成的人物侧脸照片，然后使用形变等工具调整图片，使纹理能贴合映射出来的面部模型，如图 14-55 所示。绘制完成后取消映射背景的图层显示，将文件存储为 PNG

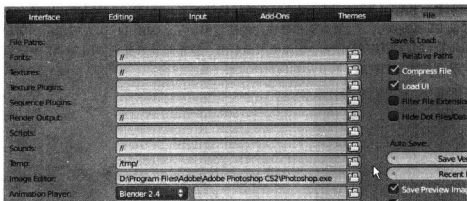


图 14-53 设置 Image Editor (图片编辑器) 的路径

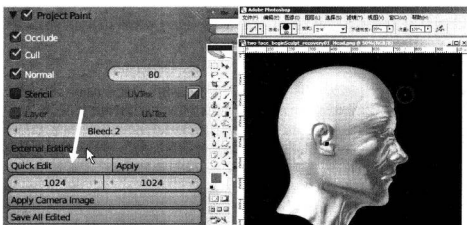


图 14-54 将当前场景画面映射至 Photoshop 中

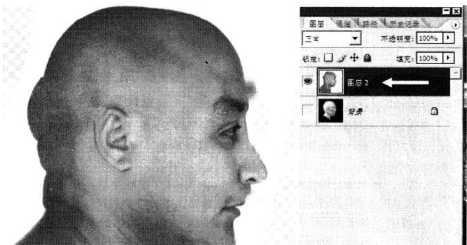


图 14-55 在 Photoshop 中绘制贴图并输出为 PNG 文件

格式,并保持原始输出的文件名。

切换回 Blender,单击 Project Paint 面板中的 Apply (应用)按钮,系统将立即自动载入刚才在 Photoshop 中绘制的贴图纹理,并将其映射至物体表面上,效果如图 14-56 所示。

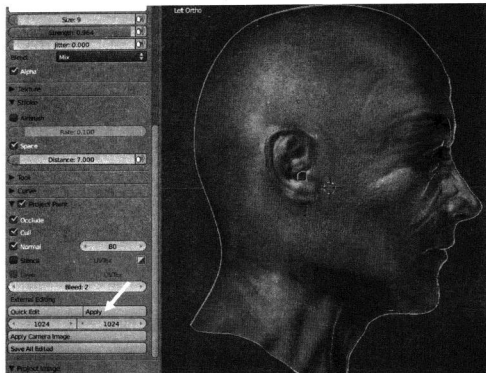


图 14-56 在 Blender 中将贴图重映射至物体表面

一般来说，只需要绘制左视图、右视图和正视图三个角度即可。通过旋转模型视图，将物体不同角度的画面都分别导入 Photoshop 中进行色彩绘制，绘制的原则应基于雕刻细节。最后的映射结果将按照模型的 UV 坐标自动展开，并保存至刚开始新建的 UV 贴图文件中。在 UV 编辑器中单击 Image（图片）菜单，选择 Save（保存）即可将映射到 UV 坐标的贴图保存至本地，如图 14-57 所示。

由于物体表面的纹理凹凸不平，所以难免会在一部分区域的映射中，因为无法接受到贴图的投影，导致最后的纹理输出上出现白色的裂纹。在 Photoshop 中重新打开刚才保存的贴图文件，使用画笔和图章等工具，修复贴图映射中出现的破损面和裂缝。

最后，需要导入刚才使用烘焙制作好的 AO 贴图，放置在色彩纹理图层之上，并选择图层模式为正片叠底，如图 14-58 所示。这样就可以借助实际纹理的 AO 光影图，来强化贴图的表面阴影，使纹理的效果更贴合雕刻细节，更富有立体感。

回到 Blender 视图中，在 UV 编辑器里单击 Reload 功能重载纹理图片，即可在 3D 视图中实时地查看最终贴图绘制的效果了，如图 14-59 所示。

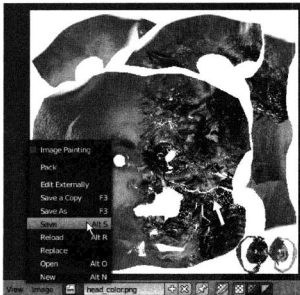


图 14-57 保存映射展开后的贴图文件

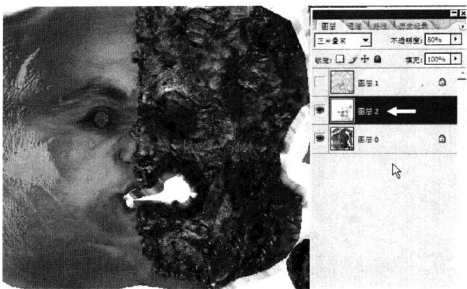


图 14-58 在 Photoshop 中完成色彩贴图的修复制作

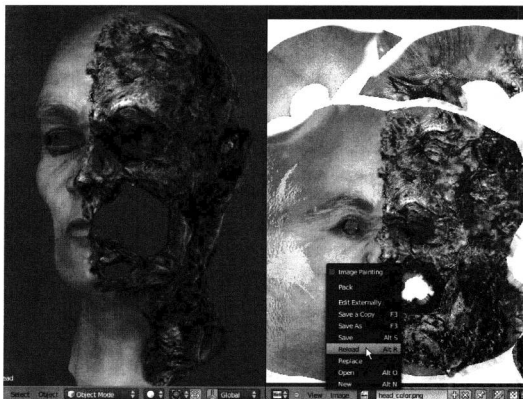


图 14-59 在 Blender 中实时查看贴图效果

完成了 Diffuse (漫反射) 贴图的绘制, 还需要制作物体的高光贴图, 用于在渲染中强调表面的高光细节。高光贴图的绘制办法有多种, 可以直接在贴图表面上手工绘制出预计存在高光效果的区域, 也可以采用接下来介绍的方法快速地制作高光贴图。

在 Photoshop 中打开刚才烘焙的 AO 贴图，使用反向工具和曲线工具，将纹理调整为如图 14-60 所示的效果。其中白色部分将在渲染中被设置为高光效果，用于表现生物体的反光质感。

同理，使用类似的绘制方法，完成其余物体的漫反射贴图。如图 14-61 所示，为本例中绘制的瞳孔和口腔肌肉贴图。



图 14-60 制作高光贴图

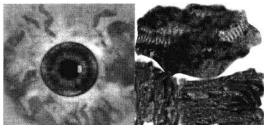


图 14-61 眼球漫反射贴图和口腔肌肉纹理贴图

## 14.8 毛发制作

首先退出编辑模式，单击 Shift + A 在头部位置添加一个 Sphere（球体），如图 14-62 左图所示。单击 Tab 进入编辑模式，切去四分之三的面，调整位置放置在右脑位置，如图 14-62 右图所示。

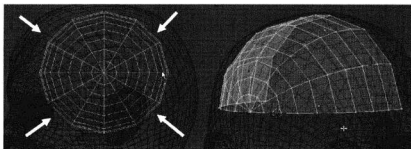


图 14-62 建立毛发射器物体

在物体模式下，转至粒子面板，为该物体新建一个粒子系统，命名为“Hair”。在粒子的 Type（类型）中选择为 Hair（毛发）类，并将 Amount（数量）设置为 500，同时减小粒子的 Normal（法向）值，效果如图 14-63 所示。

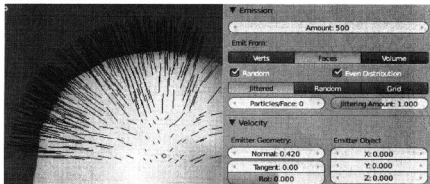


图 14-63 粒子毛发的参数设置

接着为该粒子系统添加一个 Child（子粒子）系统，如图 14-64 所示。设置属性为 Face（面）发射，调节其 Display（显示）与 Rough（粗糙度）等设置。

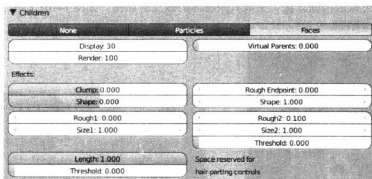


图 14-64 Children 子粒子系统的参数设置

最后设置毛发的渲染选项，开启 Strand 渲染模式，设置渲染效果为 Path（路径），如图 14-65 左图所示。完成了粒子的基本设置，就可以进入头发的梳理模式，使用 Comb（笔刷）将头发调整为如图 14-65 右图所示的发型。

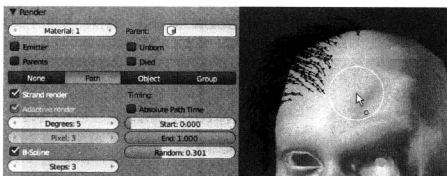


图 14-65 毛发渲染及梳理

和制作头发同理，在眉毛的部位添加一个 Plane（平面），并设置与头发相似的粒子属性，然后梳理毛发的分布，最后的效果如图 14-66 所示。

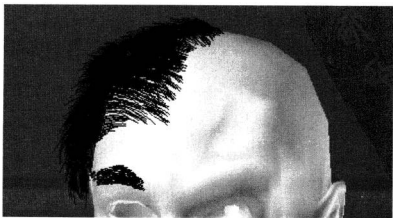


图 14-66 头发和眉毛的最终效果

## 14.9 着色器设计

首先为头部添加着色器，单击头部物体，在材质面板中新建一个材质，由于皮肤较软，所以 Hardness（硬度）值可以设置为 50，调节漫反射与高光强度值为如图 14-67 左图所示。对于具有生物特性的物体，还需要为其添加一个 SSS 着色效果，具体设置如图 14-67 右图所示。

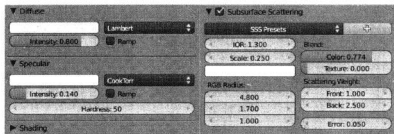


图 14-67 头部着色器的参数设置

转换至纹理面板，这里首先添加一个 Diffuse（漫反射）贴图纹理，贴图坐标选择 UV，将绘制的色彩贴图映射至 Color（色彩）通道中，如图 14-68 左图所示。接着添加一个 Spec（高光）贴图纹理，将高光贴图映射至 Spec 通道中，并适当调节输入值大小，如图 14-68 中图所示。最后添加一个 Bump（凹凸）贴图纹理，将烘焙制作的 Bump 贴图映射至 Geometry（多边形）的 Bump（凹凸）通道中，如图 14-68 右图所示。

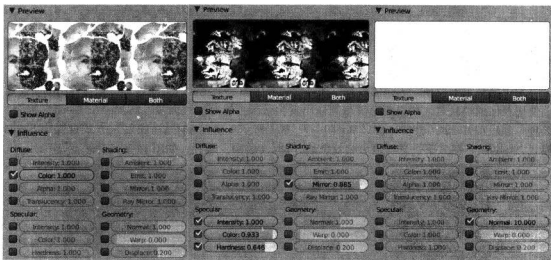


图 14-68 贴图的通道参数设置

接下来添加一个 Display（置换）贴图，将烘焙出的 Normal（法向）贴图设置在 UV 坐标系上，但是并不需要让它映射在着色器的通道中。所以这里取消贴图的可见开关，并转至物体的修改器面板，在其中添加一个 Displacement（置换）修改器，将使用了置换贴图的纹理名称输入 Texture（纹理）输入源，设置适当的 Strength（强度）值。这样就可以直接在模型上通过贴图来表现物体的高模细节了，效果如图 14-69 所示。

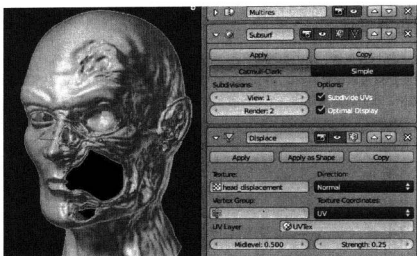


图 14-69 使用置换贴图在物体表面上显示高模细节

单击选择毛发物体，在材质面板中添加一个新的着色器，如图 14-70 左图所示，为毛发材质的基础设置面板。同时由于毛发渲染需要 Strand 模式，因此还要在 Strand 面板下，设置 Size（尺寸）的 Root（根）值和 Tip（尖）值，如图 14-70 右图所示。

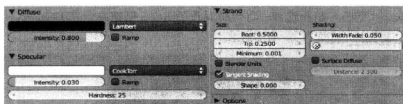


图 14-70 毛发着色器的参数设置

为了渲染出毛发顶端尖锐的效果，还需要为其添加一个线性的透明纹理，通过在 Alpha 通道上使用径向通道映射，利用渐隐纹理制作出发尖效果。转至纹理面板，添加一个 Blend（混合）纹理，并启动 Ramp（色带）图，如图 14-71 左图所示。将 Mapping（映射）方式设置为 Strand 模式，Influence 模式下设置通道影响效果是 Alpha（透明）属性，如图 14-71 右图所示。



图 14-71 设置毛发的纹理属性



眼球和嘴部肌肉的着色器设计与头部类似，这一部分的制作留给读者自己尝试。

## 14.10 灯光与环境设置

### 14.10.1 三点式照明原理

这里将采用一种三点式照明来完成场景的灯光布置。

三点式照明又称为三角形照明，它一般由三组光源组成，分别为 Key Light（主光源）、Fill Light（辅光源）以及 Back Light（背光源）。三点式照明是摄像师们在长期工作和实践中，总结出的一套灯光照明方法。

Key Light（主光源）是场景的基础光源，也是亮度最高的灯光光源。它主要控制着场面的气氛，让观众可以清楚地了解画面中主要的光源方向，同时也担负了投射出主要阴影方向的工作。主光源的位置应该与镜头和拍摄对象之间成45度左右的角，通常处于镜头与对象之间的中点位置，同时水平面上应放置在拍摄对象的顶部前上方。在室外场景中，主光源通常选用的是日光，而在室内场景中，常使用门窗照射进来的光线光源。

Fill Light（辅光源）用于平衡主光源的效果，照亮主光源没有能够覆盖到的暗部区域，并控制场景中最亮的区域和最暗的区域之间的对比度。辅光源应该处在用于照亮拍摄对象的主光源对面，用于柔化和填充主光源产生的阴影。

Back Light（背光源）用于将物体从背景中凸显出来，表现出物体的轮廓线条，常被放置在对象的背后。背光源可以是集中的，也可以是分散的。如图14-72所示，左图为一个标准三点式照明的顶视图，而右图则是从侧面观察的三点式照明光源排列。

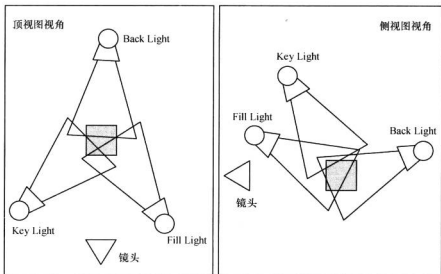


图 14-72 三点式照明

### 14.10.2 设计场景灯光和环境

在完成头部的建模后，还可以添加西装和领带等物体，具体的制作流程由读者来完成。

首先调整摄像机的角度和位置，并在镜头上方新建一盏 Area（面）光灯作为主光源，调整光源的尺寸，并设置照明区域为物体的左脸区域，照明方向是由上至下的45度角，灯光位置如图14-73左图所示。由于是整个场景的主光源，因此需要开启它的阴影范围计算，并设置合理的

强度值。如图 14-73 右图所示，为主光源的参数设置。

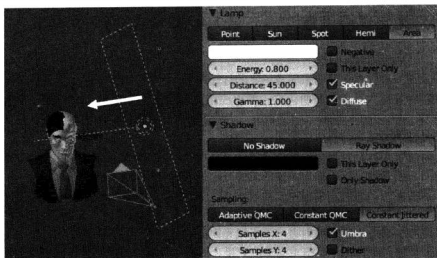


图 14-73 主光源的位置与参数设置

在右脸的斜前方，再添加一盏 Area 面光灯作为辅光源，用于照亮另外一半脸的轮廓和高光。如图 14-74 左图所示，为灯光的照明方向和尺寸调节效果。为了营造人物双面的性格特性，可以将灯光的色彩设置为蓝色。如图 14-74 右图所示，为辅光灯的参数设置。

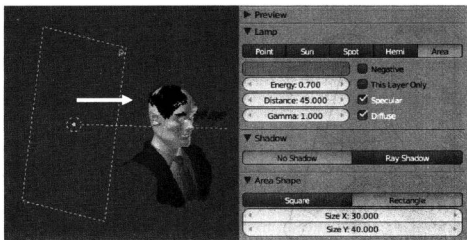


图 14-74 辅光源的位置与参数设置

为了强化左脸的边缘线条，提升表面的高光反射效果，在左脑后方添加一盏 Area（面）光源，作为背光源。如图 14-75 左图所示，为背光灯的位置和照明指向。为了突出戏剧化的色彩，可以将灯光颜色调整为绿色，如图 14-75 右图所示，为灯光的参数设置。由于只是一盏背光灯，所以不需要开启其阴影的计算属性。

转至环境面板，开启 Ambient Occlusion（环境运算），设置强度系数值为 0.8，采用 Multiply（叠加）模式。在下方的 Gather 详细设置中，使用精度较高的 Raytrace（光线追踪）算法，调整采样模式和采样率，具体设置如图 14-76 所示。

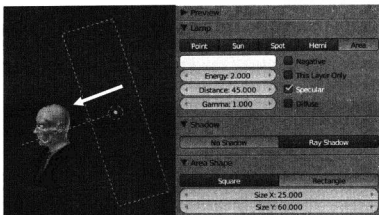


图 14-75 背光源的位置与参数设置

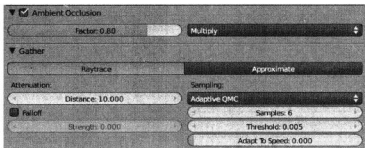


图 14-76 Ambient Occlusion (环境变量) 设置

## 14.11 结点合成

在渲染之前，需要开启一些特定的图层输出，如图 14-77 所示。在渲染的图层输出面板中，激活 Specular（高光）输出和 AO 环境运算输出。



图 14-77 输出用于后期合成的特定图层

在 Post Processing（后期处理）面板中，开启 Compositing（合成）选项，具体参数如图 14-78 所示。首先降低画面的 Resolution（分辨率）和 Anti-Aliasing（抗锯齿）输出选项，做调试渲染。

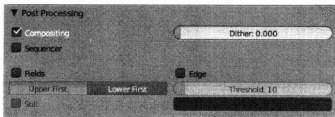


图 14-78 开启后期合成功能

单击快捷键 F12, 开始渲染原始效果图。完成后, 切换到 Compositing (合成) 工作窗口视图, 并进入 Node Editor (结点编辑器) 面板, 单击输出处理模式, 就可以在背景中看到渲染的结果了, 如图 14-79 所示。从输出预览中可以发现, 没有经过任何后期处理的渲染效果, 整体画面平淡, 人物面部没有突出的重点和细节的立体感。接下来就简单地讲解如何使用一些后期结点, 调节并处理画面的质感。



图 14-79 未经后期处理的渲染结果

首先在色彩输出接口处添加一个 RGB 曲线结点,并提高曲率以增强画面中色彩的亮度值。接着再添加一个 Blur (模糊) 结点,选择模糊模式为 Fast Gaussian (快速高斯) 模式,用于将色彩区域模糊扩散化,如图 14-80 所示。最后使用 Mix 结点,将模糊后的色彩值和原色彩输出做混合操作,这里修改模式为 Screen (屏幕) 模式,调整系数值为 0.2,这样模糊的色彩就可以对画面产生一定的柔光照明效果。

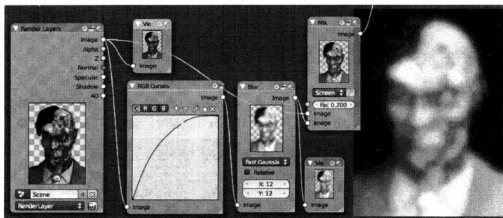


图 14-80 使用模糊结点来添加柔光效果

如图 14-81 所示, 左图为未添加柔光的画面效果, 右图为添加柔光后的画面效果, 可以看到

画面产生了一种朦胧的柔光感，并且亮度被整体提高。

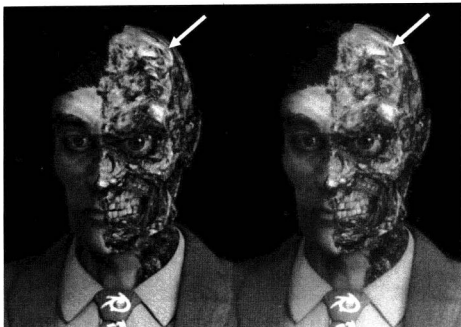


图 14-81 对比添加柔光前后的效果

接着将渲染图层的 AO 图层输出至一个 RGB 曲线结点中，调整曲线用于增强画面的阴影对比度。接着添加一个 Mix 结点，修改模式为 Multiply（叠加），将调整后的阴影叠加至刚才的色彩输出上，如图 14-82 所示。为了强调脸部上的阴影效果，可以使用一个较大的系数值 0.85。

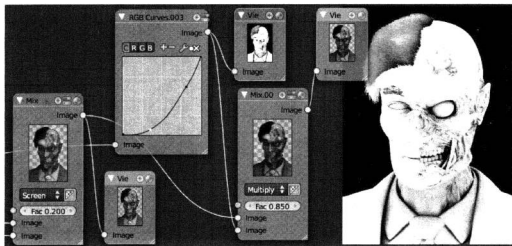


图 14-82 使用 AO 图层来增加阴影的对比度

如图 14-83 左图所示，为未叠加强化阴影的效果，如图 14-83 右图所示，为叠加 AO 后的输出效果，可以看到，画面的轮廓感和凹凸的立体感都得到了一定程度的提升。

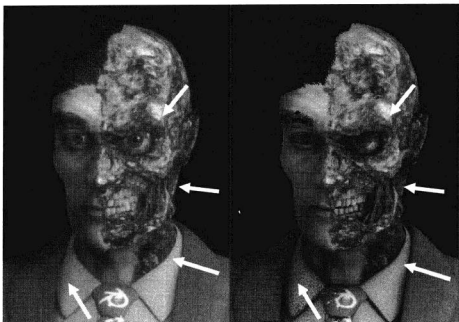


图 14-83 对比叠加 AO 前后的阴影效果

接着将图层的 Specular（高光）输出连接至一个 RGB 曲线结点中，调节曲线曲率，强化其高光值的亮度，如图 14-84 所示。接着添加一个 Mix 结点，修改模式为 Add（添加）模式，将高光的效果附加到上一层的色彩输出中。

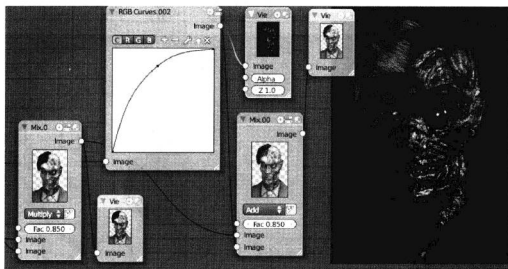


图 14-84 使用 Specular 图层输出物体的表面高光效果

如图 14-85 左图所示，为未叠加高光的效果，如图 14-85 右图所示，为叠加后的效果。可以看到，头部的高光部位主要分布在绘制的高光贴图区域，以及灯光的高光照明范围内。经处理后的高光效果强调了物体表面的生物质感，使 SSS 效果看上去更真实明显。

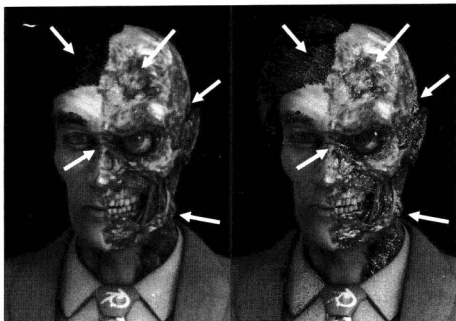


图 14-85 对比叠加强光图层前后的物体表面质感

接下来在图层输出上添加一个 Lens 特效结点，调整 Dispersion 参数为 0.02，为画面添加镜头的边缘效果，如图 14-86 所示。

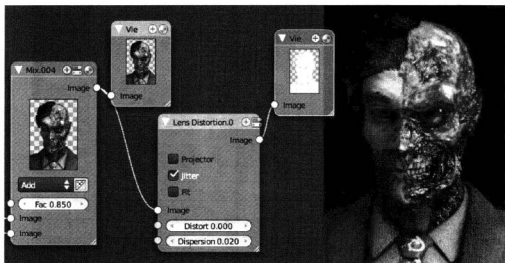


图 14-86 添加 Lens 镜头特效结点

将渲染图层的 Alpha 图层输入至一个 Blur（模糊）结点中，对其边缘稍作模糊处理，效果如图 14-87 所示。使用了一个单独的图片来作为背景图，所以这里再添加一个 Mix 结点，用于连接人物和背景图，并将刚才通过模糊处理过的 Alpha 图层，接入该 Mix 结点中的 Fac 接口，使人物和背景可以按照该 Alpha 透明区域混合起来。

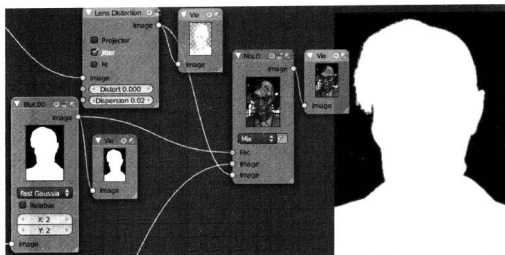


图 14-87 使用优化后的 Alpha 通道来融合人物与背景

如图 14-88 左图所示，为未使用处理过的 Alpha 通道融合人物与背景的效果，如图 14-88 右图所示，为使用经过模糊处理后的 Alpha 通道参与融合的效果。可以看到，使用模糊过的 Alpha 通道最大的作用效果，就是能羽化物体的边缘，让人物与背景的融合显得更自然。对于更具有层次感的场景，还可以尝试采用 Z 通道来完成前景与背景的融合。



图 14-88 对比使用不同 Alpha 图层融合的人物与背景效果

最后，添加一个 Color Balance（色彩平衡）结点用于调色处理，分别通过调整 Gamma 和 Gain 值来控制整体画面的色彩和氛围。由于前面的调整降低了画面的色彩亮度，所以可以在最后面再添加了一个 RGB 曲线结点，用来完成对画面亮度值的最后调整，如图 14-89 所示。完成了调节结点设计之后，就可以在输出末端添加一个 Composite（合成）结点和一个 File Output



(文件输出) 结点了。

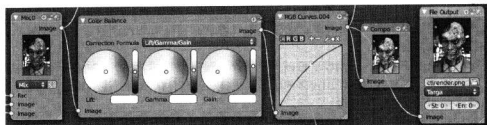


图 14-89 对画面做整体的调色处理

待完成了所有结点的设置和调整, 就可以提高画面的抗锯齿等级和 AO 的采样率等参数设置, 并取消对分辨率的缩放, 单击快捷键 F12, 做成品渲染输出, 最后的效果如图 14-90 所示。



图 14-90 Two Face (双面人)

## 14.12 本章小结

本章所介绍的工作流程仅为一个简单的角色制作演示, 在实际的人物或生物体制作时, 不宜拘束于常规的制作模式, 应根据需求来做相应的调整。接下来小结一下本章学习中提到的关键点, 以及读者需要掌握和熟练的技巧。

□ 注意生物体的表面布线规律, 应根据需求设计不同类型的布线, 特别是动画类人物。

- ☐ 合理地使用 Loop 结构来模拟制作关键肌肉。
- ☐ 减少三星点和五星点在关键位置上的使用，因为它们将影响人物的动作和表情制作。
- ☐ 拆分和调整 UV 时，一定要多在视图中实时地查看调整后的贴图效果，及时纠正表面上纹理的拉伸错误。
- ☐ 雕刻时应由简入深，在低模下可以勇于修改，做出大胆的尝试。在高模下要合理把握细节的深入，不要盲目地使用过多的细节，去掩盖掉整体的结构，注重画面的张弛有度。
- ☐ 掌握基本的贴图烘焙技术。
- ☐ 掌握映射绘制技术，能配合第三方软件来制作 Diffuse 色彩等贴图。
- ☐ 掌握三点式照明布局。
- ☐ 掌握常用的结点处理流程。

## 室内效果图应用

本章将讲解如何结合开源渲染器 Yaf(a)ray 和 Blender 来制作简单的室内效果图。

建筑和室内效果图是 3D 应用中较广泛的领域之一，许多地产商和家装设计师都会借助于 3D 来制作预览效果图。本章中将简要讲解建模思路和流程，重点讨论 Yaf(a)ray 中的灯光、材质和环境等渲染参数设置。

## 15.1 场景建模

### 15.1.1 制作桌子

首先制作一张桌子，单击小键盘的 Num7 进入顶视图视角，使用 Shift + A 添加一个默认 Circle 物体，如图 15-1 左图所示。接着旋转视图，按 A 全选所有的点，单击 E + G + Z 向 Z 负轴挤压一小段距离，然后单击 S 向内缩小一部分，制作出桌子的边缘，如图 15-1 右图所示。

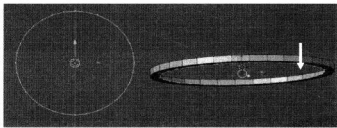


图 15-1 添加一个默认 Circle 物体制作桌面

继续使用 E + G + Z 向 Z 轴的负方向垂直挤压，如图 15-2 左图所示，完成桌面的厚度制作。接着转向桌面，使用 Alt + RMB 选择最上方的边，单击快捷键 E，然后松开鼠标，接着再单击 Alt + M 将刚才新挤压出的点合并并在圆心处位置，如图 15-2 右图所示。

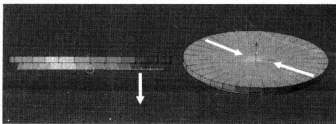


图 15-2 挤压桌面

选择桌子下方的 Circle 圈，单击 E + S 向内缩小挤压面，然后再按 E + G + Z 向下挤压出中柱部分。接着在修改器面板中，为当前物体添加一个 Subdivision Surface（细分修改器）。可以看到，由于边缘的面数较少，所以桌体的转折边变得十分圆滑，如图 15-3 所示。

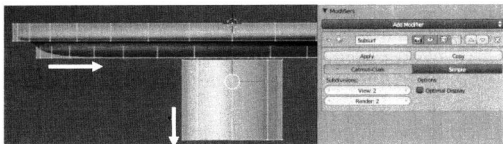


图 15-3 添加细分修改器后的预览效果

先单击修改器上的“小眼睛”，关闭细分预览，继续完成模型的制作。使用 Alt + RMB 选择桌底的边并继续向下挤压，延长桌子的中柱长度，如图 15-4 左图所示。接着使用 E + G + Z + S 制作出中柱上的球面体，如图 15-4 中图所示。最后在末端使用 Alt + M 合并所有的点至圆中心位置，如图 15-4 右图所示。

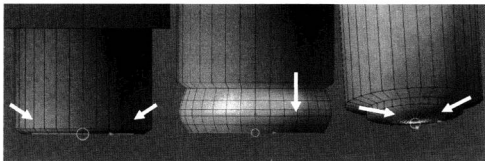


图 15-4 完成桌柱制作

将光标移动至桌面下方，单击 Shift + A 添加一个 Cube 物体，如图 15-5 左图所示。使用 Ctrl + R 做环切细分，并使用 G 调整切割线的位置和角度，最后的效果如图 15-5 中图所示。单击小键盘的 Num7 转至顶视图视角，使用 LMB 将光标定位在桌柱中心，单击 T 调出工具菜单，选择 Spin（旋转）工具，并在 Steps（旋转步长）选项中输入 5，这样桌腿将围绕光标自动复制 5 次，效果如图 15-5 右图所示。

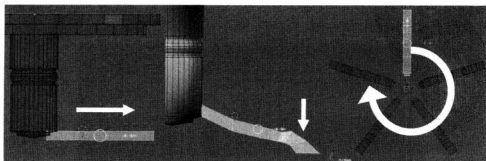


图 15-5 制作桌腿

完成了桌面的基本建模,下面打开细分修改器上的小眼睛,激活预览模式。使用第13章中介绍的硬边制作方式,单击 Ctrl + R 在各侧面位置添加 Loop 切割线,并将切割线放置在靠近转折边缘的位置,如图 15-6 左图所示。以此类推,对桌子所有的边都执行硬边处理,最后的调整效果如图 15-6 右图所示。

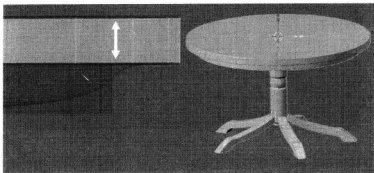


图 15-6 处理硬边效果

### 15.1.2 制作椅子

接下来制作椅子。单击小键盘的 Num7 转换到顶视图视角,按住 Shift + A 添加一个 Plane 物体,并在修改器面板中为其添加一个 Mirror (镜像) 修改器,如图 15-7 左图所示。接着使用 Ctrl + R 对其做切割细分,调节效果如图 15-7 右图所示。

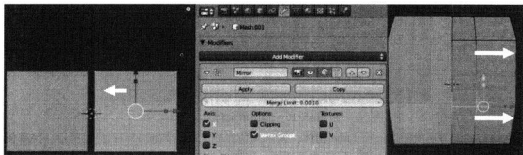


图 15-7 制作椅面

使用快捷键 Alt + RMB 选择平面的外周边,单击 E + Z 向下挤压一定距离,制作出椅子的厚度,接着再单击 E + S 向内挤压,效果如图 15-8 左图所示。接着继续单击 E + Z 向下挤压,完成椅面的制作,如图 15-8 右图所示。

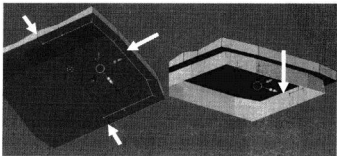


图 15-8 挤压椅面

使用制作桌腿的方法,在椅面的下方角落位置添加一个 Cube 物体,如图 15-9 左图所示。使

用快捷键 S 缩小其大小，并选择底面，单击 G + Z 向 Z 轴下方做挤压操作，制作出椅腿。接着在椅腿的中间位置添加一个切割线，为其增加一点弧度，完成后的椅腿效果如图 15-9 右图所示。

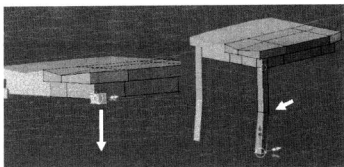


图 15-9 制作椅腿

单击 Shift + D 将椅腿复制后再按 G 移动到椅子的后面，如图 15-10 左图所示。选择最上面的面，单击 E + Z 向上挤压，延长后面椅腿的高度，如图 15-10 右图所示。

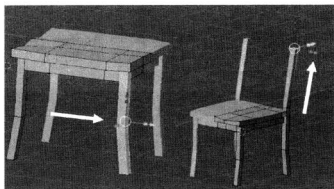


图 15-10 制作剩余椅腿

在椅背的上方再次添加一个 Cube 物体，调整其位置和大小，如图 15-11 左图所示。接着使用 Ctrl + R 切割出靠背的形状，如图 15-11 中图所示。同理，在椅背正下方的位置，单击 Shift + A 添加一个 Cube 物体，使用 S 做缩放操作后，利用 Ctrl + R 切割出椅背靠杆的形状，如图 15-11 右图所示。

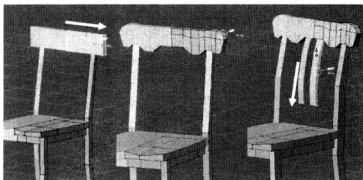


图 15-11 制作靠背

在椅背靠杆的下方添加一个 Cube 物体，完成靠背的全部结构制作，如图 15-12 左图所示。接着在修改器面板中为其添加一个 Subsurf（细分）修改器，如图 15-12 中图所示。利用制作桌子硬边的技术对其做硬边处理，得到了椅子的最终效果，如图 15-12 右图所示。

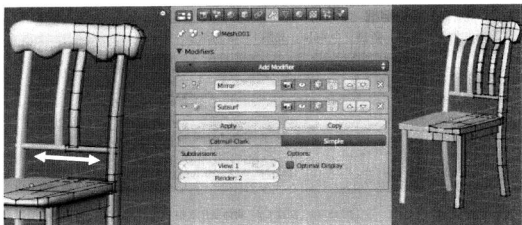


图 15-12 对椅子做硬边处理

按 Tab 回到物体模式，选择椅子并使用 Alt + D 进行关联复制，将椅子摆放在桌子的四周，如图 15-13 左图所示。由于属性的关联性，这样新复制出的椅子会和原椅子做同步的修改与更新。桌子上可以添加一个盘子，制作方法很简单，只需要按住 Shift + A 新添加一个 UV Sphere，然后删除一半的面，将剩下的面全选 S + Z 做压缩即可。接着在盘子里随机摆上一些大圣物体做装饰，如图 15-13 右图所示。

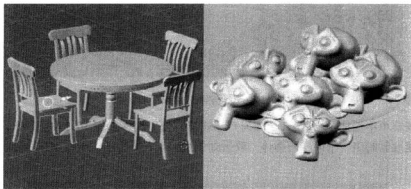


图 15-13 摆放座椅位置及制作盘子

在物体模式下按住 Shift + A 新添加一个 Plane 物体，使用 S + Z 做垂直缩放，做出一个简单的地毯，并按 G 将其摆放到座椅下方，如图 15-14 所示。

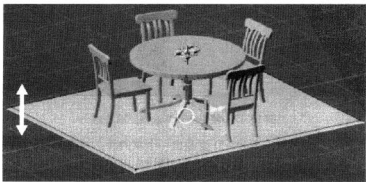


图 15-14 制作地毯

### 15.1.3 制作衣柜和相框

接着来制作一个衣柜，新建一个 Cube，如图 15-15 所示。



图 15-15 新建 Cube 物体制作衣柜

单击 Tab 进入编辑模式，使用 S + X 做水平缩放，如图 15-16 所示。

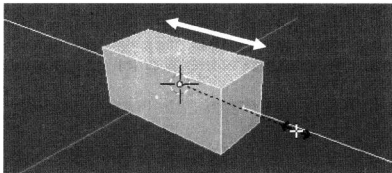


图 15-16 水平缩放 Cube 物体

在修改器菜单中添加一个 Array（阵列）修改器，设置属性如图 15-17 左图所示。利用修改器，可以快速地复制出 3 个同样的物体，效果如图 15-17 右图所示。最后单击 Apply（应用）修改器的效果。

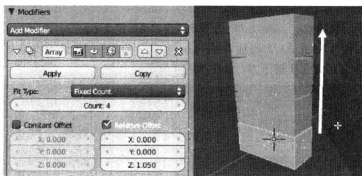


图 15-17 使用阵列修改器复制物体

选择物体侧面的一个面，单击 Shift + D 复制出一个新的面，如图 15-18 左图所示。接着单击 S + Z 做垂直缩放，并挤压出一个体。最后的效果如图 15-18 右图所示。



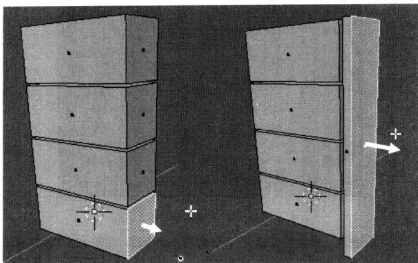


图 15-18 制作柜子的侧面

同理制作出柜子的顶，并挤压出一个体，如图 15-19 所示。

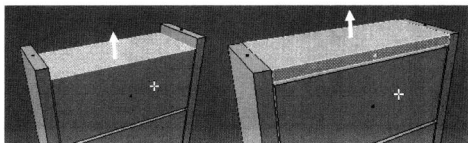


图 15-19 挤压出柜顶

接着再添加一个小 Cube 物体，单击 S 缩小其尺寸，如图 15-20 左图所示。转换至边选择模式，选择最外面的两个边，单击 S+Z 做垂直缩放，完成抽屉把手的制作，如图 15-20 右图所示。

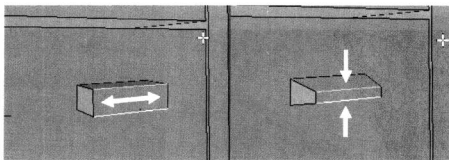


图 15-20 制作抽屉把手

将把手复制多次，最后即可得到一个简易的橱柜，如图 15-21 所示。

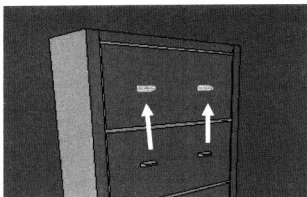


图 15-21 完成橱柜制作

再添加一个 Cube 物体，单击快捷键 S+Z 将其沿 Z 轴压缩，如图 15-22 左图所示。接着选择顶上的面，单击 S+Z 对新复制出来的面做缩放操作，如图 15-22 中图所示。最后单击 S+G+Z 将这个面沿 Z 轴向下做缩放操作，并挤压至一定的厚度，如图 15-22 右图所示。一个相框就算制作好了。

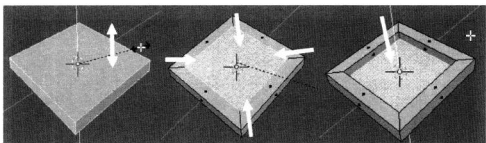


图 15-22 制作相框

#### 15.1.4 制作地板和墙体

接下来开始制作地板，首先新建一个 Cube，单击快捷键 S+Z 在 Z 轴上做压缩操作，用于制作地板砖，如图 15-23 左图所示。接着在修改器选项中为其添加 2 个阵列修改器，分别用来复制 Y 轴和 X 轴上的地板砖数量，其中 X 轴阵列修改器的参数设置，如图 15-23 右图所示。

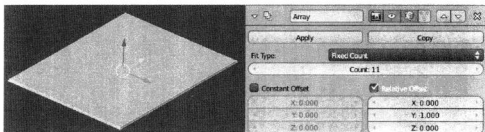


图 15-23 制作地板砖

最后调节地板砖之间的位置，产生交错的效果，如图 15-24 所示。

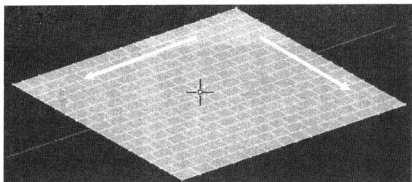


图 15-24 地板成品

添加一个 Plane (平面), 使用 S 缩放至地板大小, 再单击快捷键 E + Z 向 Z 轴正方向挤压, 制作出墙壁, 如图 15-25 所示。

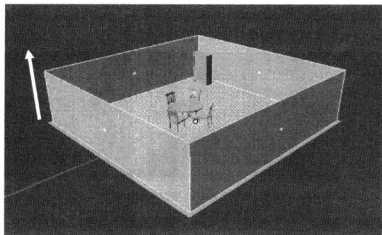


图 15-25 制作墙壁

选择旁边的一个面, 单击 E + S 做挤压缩放操作, 如图 15-26 左图所示。可以配合快捷键 Y 在缩放时约束横向变形, 制作出一个正方形的门。最后单击 X, 删除这个面, 即可完成门的制作, 如图 15-26 右图所示。这个门本身并不会出现在场景渲染中, 制作门只是为了模拟某一个方向上的光线, 用于为场景添加更多令人信服的光照细节。

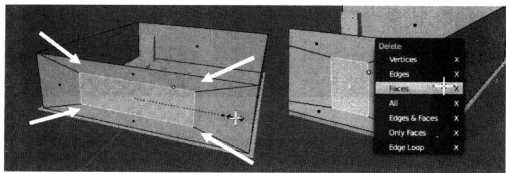


图 15-26 制作门

接着使用同样的方法，在另一侧的墙上挖出两个面，制作出窗户，效果如图 15-27 所示。

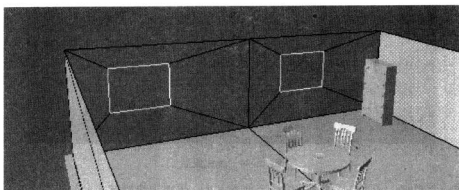


图 15-27 制作窗户

在修改器中为墙体添加一个 Solidify（实体化）修改器，如图 15-28 左图所示。调节 Thickness（厚度）值，完成墙体的厚度制作，效果如图 15-28 右图所示。

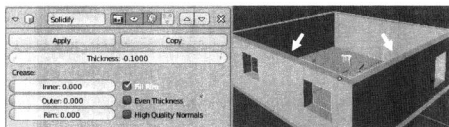


图 15-28 使用修改器制作墙体厚度

接着制作墙体的踢脚线，首先添加一个 Cube，使用 S 缩小尺寸，如图 15-29 左图所示。选择侧面的面，单击 S+Z 在 Z 轴上稍微压缩一定比例，如图 15-29 右图所示。

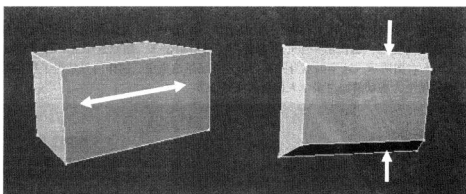


图 15-29 制作踢脚线

单击 S 对踢脚线做横向缩放，并使用快捷键 G 将其安装至墙体底部，同时再添加一个 Cube 物体，制作出窗台，效果如图 15-30 所示。

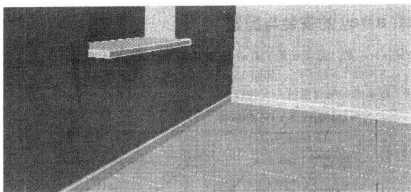


图 15-30 安装踢脚线和窗台

最后添加一个 Cube 物体, 使用快捷键 S + Z 做缩放操作, 制作一个屋顶, 并放置在如图 15-31 所示位置。

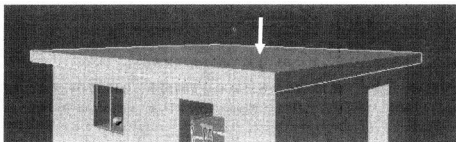


图 15-31 制作屋顶

放置摄像机, 使用小键盘 Num0 切换至镜头视角, 将刚才制作的相框摆放至窗台上, 调节各物体的位置, 最后的效果如图 15-32 所示。

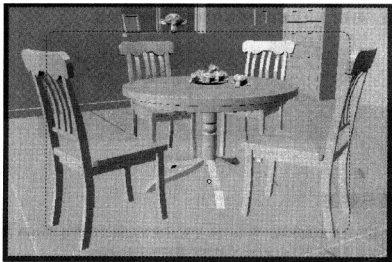


图 15-32 完成场景的搭建

## 15.2 Yaf(a)ray 的安装与配置

在使用 Yaf(a)ray 前, 需要安装 Yaf(a)ray 渲染器并配置其输出脚本。

进入它的官方网站<sup>①</sup>下载专区, 如图 15-33 所示, 可以找到最新版的 Yaf(a)ray 渲染器安装程序, 选择并下载适合的操作系统的安装版本。

**YafaRay 0.1.1 Download.**

Please read these **installation notes** before installing. Use the forum if you have any issues about installing YafaRay.

Operating System	Download	Release Notes	Filetype	Size	Version
Windows 32 bits	<b>YafaRay 0.1.1 - Python 2.6</b> (for Blender releases built with Python 2.6)	0.1.1	exe Installer	6.0 MB	0.1.1
<b>Notes for Win32:</b> YafaRay 0.1.1 needs <b>Python 2.6.2</b> installed. It will only work with Blender versions compiled with <b>Python 2.6</b> . In <b>Windows 64 bits</b> systems, YafaRay 0.1.1 should be used with Windows 32 bits versions of Blender and Python. Read these notes if you experience any problem.					
Mac OS X Intel 10.5	<b>YafaRay 0.1.1 - Python 2.5.1 Apple</b> (Thanks to Jens Verweide)	0.1.1	zip Installer	10.7 MB	0.1.1

图 15-33 下载 Yaf(a)ray 安装程序

完成安装后, 还需要下载 Blender 的导出脚本与界面脚本, 如图 15-34 所示, 选择并下载 Yaf(a)ray 的 UI 脚本, 将其解压后拷贝至 Blender 的脚本目录 \ scripts \ ui 中即可。

	YafaRay	0.1.1	zip file	530 KB	0.1.1
<input type="checkbox"/> Sources	Blender export scripts	0.0.1	zip file	38 KB	0.1.1
For installing YafaRay from sources, please read this page					

图 15-34 下载 Blender 导出脚本

最后运行 Blender, 即可在渲染器选项菜单中找到外置的 Yaf(a)ray 选项了, 如图 15-35 所示。单击选择后, 系统将会自动转换渲染和材质等参数的面板设置。

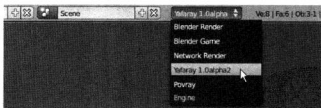


图 15-35 选择渲染器

## 15.3 材质与贴图

由于物体的着色计算原理不同, Yaf(a)ray 中对材质和贴图的设置与 Blender 内置的设置不太一样, 所以这里需要重新认识 Yaf(a)ray 中对着色器和贴图设置。

① 网站地址: [www.yafaray.org/download](http://www.yafaray.org/download)。

### 15.3.1 着色器

Yaf(a)ray 提供了 4 种着色器类型。切换至材质选项中, 首先为任意物体添加一个材质着色器, 如图 15-36 所示。物体的材质面板将自动转换成 Yaf(a)ray 的着色器设置界面。



图 15-36 Yaf(a)ray 的着色器参数面板

合理利用预览功能可以帮助快速而高效地完成对着色器的调节, 这里一共提供了两种材质的预览模式。其中, Preview Material 仅预览当前的材质效果; 而 Preview Object 可结合默认的灯光设置, 来预览当前物体在应用材质后的渲染效果。

材质面板中提供的 4 种材质类型, 涵盖了大部分常见物体的质感效果, 可以总结不同着色器类型常用于制作的物体类型如下。

- ☐ Shiny: 漫反射类着色器, 常用于制作大部分漫反射类物体, 例如石头, 锈金属, 混凝土, 布料, 纸张, 布匹, 石膏, 木料, 窗帘, 发光物体, 镜面, 透明或半透明属性的材料等。
- ☐ Glossy: 光泽类着色器, 可用于制作所有类型的塑料材质, 还包括亮洁的金属, 打蜡的木材, 有机材料的表面等。
- ☐ CGlossy: 用于制作多涂层的物体表面, 例如车身烤漆等。
- ☐ Glass: 玻璃类的专用着色器, 同时也可用于制作水等液体。

#### 15.3.1.1 Shiny (漫反射) 类着色器

Shiny (漫反射) 类着色器是最基本的材质着色器, 如图 15-37 所示, 为 Shiny 材质的参数面板。

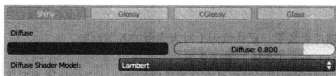


图 15-37 Shiny 材质的参数面板

漫反射是当光线照射在物体表面上后, 向各个方向反射的现象。表面漫反射的颜色和明暗度, 分别由拾色器和 Diffuse 值来控制。同时 Yaf(a)ray 还提供了多种漫反射的着色器类型, 如 Lambert 等, 其原理和区别与 Blender 的内置着色器类似, 读者可以参考前面的材质章节, 理解并比较。如图 15-38 所示, 为使用一个标准漫反射材质制作的石料渲染效果。

如图 15-39 所示, 为 Shiny 的 Specular (高光) 和 Mirror (镜面) 反射的参数面板。

**Specular:** 高光的颜色和强度值设置。

**IOR:** 镜面折射率。

**Emit:** 物体的材质自发光选项, 自发光的光线色彩

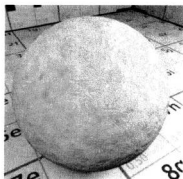


图 15-38 Diffuse (漫反射) 着色的渲染效果

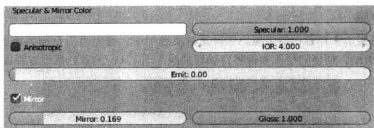


图 15-39 Specular (高光) 和 Mirror (镜面) 反射的参数面板

由 Diffuse (漫反射) 的色彩决定。如图 15-40 所示, 为使用 Emit (自发光) 着色材质的物体效果。

**Mirror:** 该参数是高光反射与漫反射之间的混合比例, 值越高, 则高光反射越强, 漫反射值越低。同时配合 IOR 的值可调整着色时的菲涅尔强度效果, 在同样 Mirror 值的情况下, IOR 值越高, 则反射强度也越高。如图 15-41 所示, 为不同 Mirror Strength (强度值) 与 IOR 值组合的材质渲染效果。



图 15-40 材质 Emit (自发光) 的渲染效果

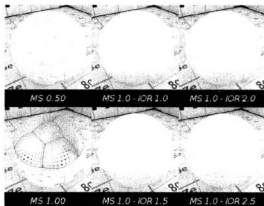


图 15-41 不同 Mirror Strength (强度值) 与 IOR 值组合的材质渲染效果

**Gloss:** 添加模糊反射效果, 默认值为 1 时为光滑的表面反射。

### 15.3.1.2 Glossy 和 CGlossy (光泽) 类着色器

Glossy (光泽) 类材质常用于制作具有金属质感的物体, 例如金属器皿等。但是 Glossy 材质的一个缺陷就是它在高光处容易产生噪点, 解决办法只有调高灯光的采样率, 但是这需要花费更多的渲染时间。如图 15-42 左图所示, 为灯光采样率为 2 时, 反射高光位置的噪点效果, 而右图则是采样率为 64 的效果。

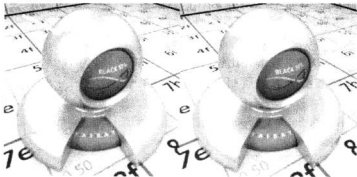


图 15-42 灯光采样率对物体反射光的噪点影响效果



同时, Glossy 还支持 Anisotropic (各向异性) 的反射效果, 如图 15-43 所示, 为 Anisotropic (各向异性) 反射的参数面板。

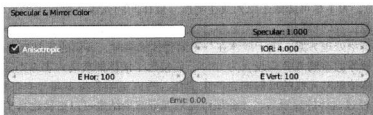


图 15-43 Anisotropic (各向异性) 反射的参数面板

前面曾介绍过, 各向异性是在表面处理中, 由于纤维或划痕造成反射沿着轴线模糊或拉伸的一种特殊金属表面反射效果, 常用来制作磨砂的金属效果等, 例如抛光的厨房用具。普通的反射光方向是朝向四面八方等量的, 而各向异性反射则控制着物体的表面反射方向。当开启 Anisotropic (各向异性) 选项后, 面板下方会出现 E Hor (Exponent Horizontal) 水平指数和 E Vert (Exponent Vertical) 垂直指数, 它们的调节方式和坐标原理与 UV 映射坐标相同。如图 15-44 所示, 左边物体为开启了各向异性的物体反射效果, 而右边则是默认的漫反射效果。

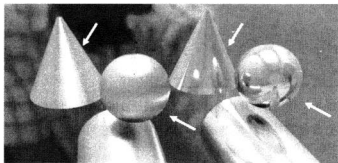


图 15-44 启用各向异性的材质高光反射效果比较

CGlossy (Coated Glossy) 是一种基于 Glossy 光泽类材质的强化类, 它看上去像是在材质的表面层又添加了一个光泽类的涂层效果。可以使用 IOR 来调节这个反射涂层的高光值, 同时控制其对焦散的影响。如图 15-45 所示, 为不同 IOR 参数值下的 CGlossy 着色器效果, 可以看到, 这个着色器特别适合作车身烤漆效果。

#### 15.3.1.3 Glass (玻璃) 类着色器

玻璃本身是一种吸收体, 它能与光线在环境中产生反射和折射效果。在 Yaf(a) ray 中, 玻璃材质只有吸收计算是需要手动设置的, 而基于物理现象的折射和色散等效果, 则由 IOR 折射率来自动控制。对于过滤器和镜面反射等物理中并不存在的属性参数, 则只是用于模拟真实效果的一种科学参数。

为了获得真实的渲染计算效果, 不仅要小心地调节着色器参数, 还需要在一些细节上多加注意。例如当制作一些液体近景或特写镜头时, 参

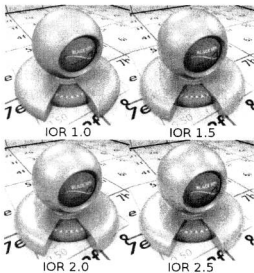


图 15-45 IOR 控制 CGlossy 的涂层反射效果

考如图 15-46 左图所示的真实物理世界，水面和物体间并没有出现完全垂直的过渡交界，这是由于自然界中液体存在一种不垂直于表面的张力，使边缘处会产生一种圆弧形的边界过渡。

在 3D 制作中，为了尽可能地还原光线在物体与液体间的光线衍射和张力等物理效果，应该在建模时，对边缘位置的模型做一些调整和处理。如图 15-46 右图所示，为一个玻璃杯盛装液体时，液体边缘的模型处理方案。由于液体在与不同物体接触时，表面的反射率也会产生一定的变化，因此未接触液体的玻璃 IOR 应设置为 1.55，而接触了液体的玻璃内表面则需要设置为 1.16。同时，空气和液体接触时光线的反射率会产生变化，所以还需要将表面与空气接触的液体反射率设置为 1.33。经过这样的设置后，才能尽可能地保证渲染结果可以更接近真实的物体效果。

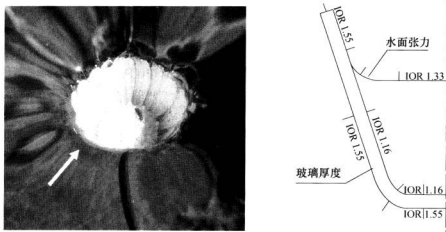


图 15-46 使用补充建模和材质设置来还原液体与物体边缘的张力效果

Glass（玻璃）类着色器是一种标准的玻璃属性材质，它可在模型的基础上模拟光线反射和透射，同时还可产生焦散效果。Caustics（焦散）是一种间接照明的形式，它是光滑物体反射的光线和通过透明物体生成的聚焦光线，在其他物体上形成的照亮区域。如图 15-47 所示，为自然界中的焦散效果。

要制作玻璃的焦散效果，常使用太阳光为主要光源，同时玻璃或液体的反射率应至少大于 1。在渲染设置中，还需要开启焦散的计算，并且焦散的光线深度值应大于物体表面的透射深度值。具体如何制作焦散效果的参数设置，将在后面的渲染设置中讨论。

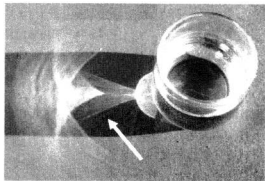


图 15-47 Caustics（焦散）效果

如图 15-48 所示，为 Glass 的 Transparency（透明度）选项。

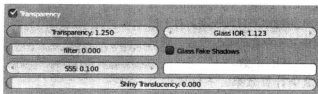


图 15-48 Transparency（透明度）选项

**Transparency:** 透明度选项, 用于设置材质的透明程度, 系统将根据物体的透明度, 自动产生对应的过滤色彩阴影。如图 15-49 所示, 为材质的透明渲染效果。

**Glass IOR:** 反射率设置, 用于调节不同物体中光线的反射和折射方向, 具体的数值选择请参考表 6-1。

**Filter:** 光线过滤值。入射光在经过一个物体时, 由于物体本身的特性, 一部分光会被吸收或者过滤掉。同时材质对光线的吸收强度, 也影响着玻璃的色彩和焦散光线的长度。因为光线前进的深度距离越深, 被吸收的光线也就越多, 所以玻璃在一定厚度的区域就会比其他厚度的位置显得更暗。

如图 15-50 所示, 为使用不同 Filter 值时玻璃对光线吸收的渲染效果。

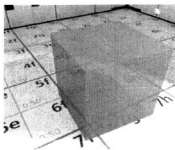


图 15-49 物体的透明效果

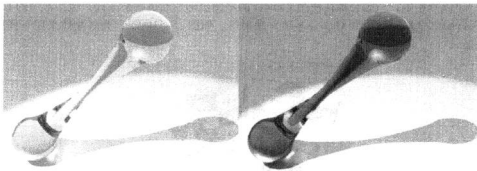


图 15-50 Filter (过滤) 值对光线的吸收效果

**Glass Fake Shadows:** 开启光线的伪阴影, 使光线追踪将按照 Filter 的值来计算阴影的色彩值。如图 15-51 所示, 请将左图与右图开启了 Fake Shadows 选项的阴影色彩做对比。

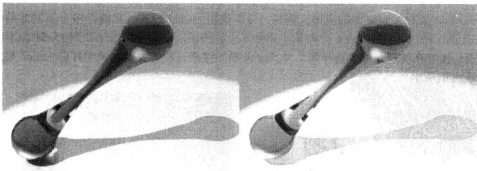


图 15-51 Glass Fake Shadows (虚拟光线) 阴影

**SSS:** 开启 SSS 效果。

**Shiny Translucency:** 开启半透明效果, 控制玻璃背后物体的色彩在玻璃表面的漫反射效果。如图 15-52 左图所示, 为现实中的物体半透明效果, 而右图则是在 3D 环境中模拟的半透明效果。

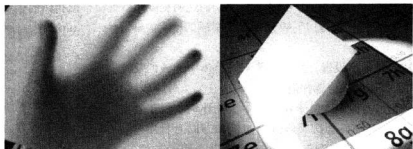


图 15-52 Shiny Translucency (半透明) 效果

## 15.3.2 纹理贴图与通道映射

### 15.3.2.1 Texture (纹理) 应用

在 Yaf(a)ray 中, 系统会默认使用 Blender 的贴图坐标和通道映射方式, 包括常用的 UV 贴图映射、Global (全局) 映射和 Orco (本地) 映射等。如图 15-53 所示, 为对物体采用不同贴图映射的效果。



图 15-53 不同贴图坐标映射方式的效果

Yaf(a)ray 支持多重贴图的应用, 例如, 可以使用 Stencil 模式, 来混合多个不同的贴图, 制作出高级的纹理效果。首先在一个材质中添加 3 个纹理图层, 并分别添加如图 15-54 左图所示的 3 个贴图。接着在纹理面板中选择第二个 Stencil 贴图纹理, 在其映射选项中开启 Stencil 模式, 如图 15-54 右图所示。

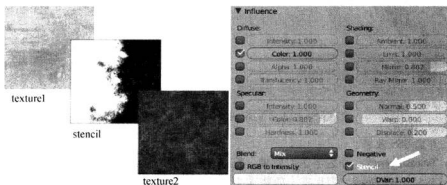


图 15-54 使用 Stencil 模式来混合多贴图

最后渲染的结果,如图 15-55 所示,Stencil 图层将上下两个图层按照 Alpha 值自然地混合了起来。

当然,也可以直接使用纹理中的混合模式,来完成纹理图层间的叠加效果。目前,Yaf(a)ray 支持的混合模式包括 Lighten (变亮), Darken (变暗), Divide (排除), Difference (差值), Screen (滤色), Multiply (正片叠加), Subtract (减去), Add (叠加) 和 Mix (混合) 模式。如图 15-56 所示,为接下来用于演示图层混合的两个原始纹理。

使用如图 15-56 左图的纹理为基础纹理,接着修改右图中的纹理混合模式。最终的渲染效果如图 15-57 所示,这里分别使用了 Add、Multiply、Difference、Lighten 四种混合模式。

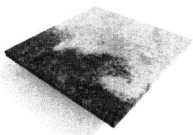


图 15-55 混合贴图的最后效果

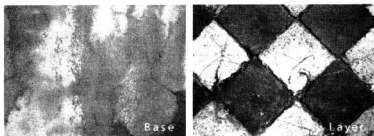


图 15-56 原始纹理

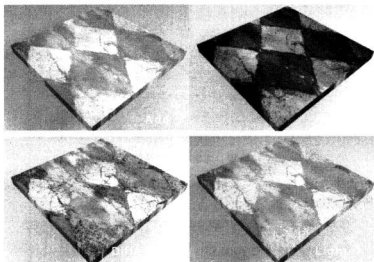


图 15-57 纹理图层的混合模式

#### 15.3.2.2 Channel (通道) 效果

对于映射通道, Yaf(a)ray 支持常用的纹理贴图通道, 包括 Color (色彩) 通道, Spec (高光) 通道, Reflec (反射光) 通道, Bump (凹凸) 通道, Displacement (置换) 通道, Alpha (透明) 通道等。如图 15-58 左图所示, 为映射纹理至 Bump (凹凸) 通道的效果, 而右图则为映射至 Displacement (置换) 通道的效果。

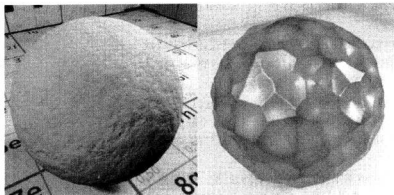


图 15-58 Bump 凹凸通道和 Displacement（置换）通道的映射效果

### 15.3.3 设计场景材质

了解了 Yaf(a)ray 的材质着色器原理, 就可以开始为场景中的物体分别添加材质了。首先选择地板, 在材质面板中为其添加一个着色器, 就可以直接在 Yaf(a)ray 属性面板中调节具体的参数数值。对于地板可以使用 Glossy 材质, 为其添加细微的反光属性。如图 15-59 所示, 为地面材质的参数设置。同时添加一个木料纹理, 建立方式与 Blender 中添加纹理的方式相同。

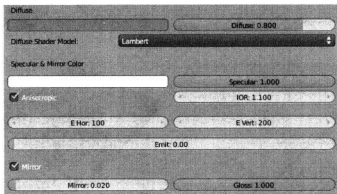


图 15-59 地板材质的参数设计

接着选择墙体物体, 设置为 Shiny 类材质, 由于墙体没有反光的效果, 所以取消高光的数值。如图 15-60 所示, 为其材质的参数设计。

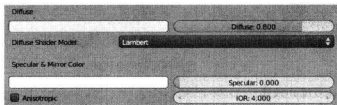


图 15-60 墙体材质的参数设计

选择踢脚线和窗台物体, 使用 Shiny 类材质, 并赋予一个木料纹理。如图 15-61 所示, 为踢脚线物体材质的参数设计。

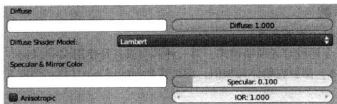


图 15-61 踢脚线材质的参数设计

选择地毯物体，使用 Shiny 类材质，并添加一个毛毯纹理。如图 15-62 所示，为地毯材质的参数设计。

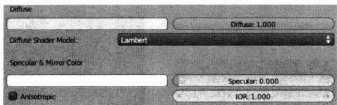


图 15-62 地毯材质的参数设计

桌椅可以使用同样的材质，设置为 Glossy 类型，赋予一个木料纹理。如图 15-63 所示，为桌椅物体材质的参数设计。

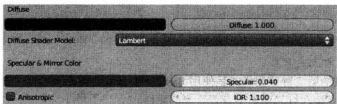


图 15-63 桌椅材质的参数设计

最后选择桌上的大圣物体，设置其类型为 Glass 类材质。如图 15-64 所示，为其透明度的参数设计。

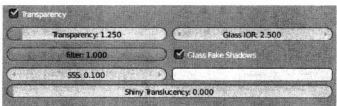


图 15-64 玻璃材质的参数设计

## 15.4 灯光和镜头

对光线追踪类渲染器，灯光设计十分重要，接下来的一节中将讨论 Yaf(a)ray 的灯光原理。

## 15.4.1 灯光参数

### 15.4.1.1 Mesh Light (物体光源)

Yaf(a)ray 提供了两种灯光照明模式, 第一种照明模式是使用 Mesh Light (物体光源)。当在场景选择一个物体, 然后在物体面板中, 即可找到 Mesh Light (物体照明) 的参数面板, 如图 15-65 所示。

当激活了物体的自发光照明选项后, 它将能成为一种面光源物体, 用于照亮周围的物体。

**Samples:** 光线采样率, 用于计算照明物体时产生的阴影边缘, 采样率越高, 阴影越平滑, 但是需要消耗更多的渲染时间。

**Power:** 光照强度, 控制物体的发光强度值。

**Radiance Color:** 用于选取物体光线的色彩, 如图 15-66 所示, 为使用物体光照明的场景效果。

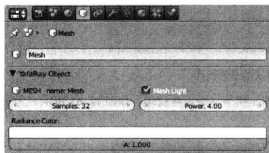


图 15-65 Mesh Light (物体照明) 的参数面板

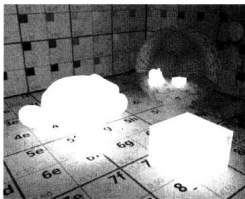


图 15-66 Mesh Light (物体光源) 的照明效果

### 15.4.1.2 Lights (灯光光源)

Yaf(a)ray 提供的第二种照明模式是使用 Blender 中自带的灯光来照明, 当选择场景中的一个灯光物体后, 可以直接在灯光面板中设置灯光的属性。

对于 Point (点光源), 是否设置灯光为 Sphere (球体) 照明, 将影响最终的照明阴影边缘。如图 15-67 所示, 左图为普通的 Point (点光源) 照明效果, 它的阴影边界是十分清晰的, 而右图是将灯光转换为 Sphere (球体) 式照明后的效果, 可以看到投射后的阴影边缘变得十分模糊。

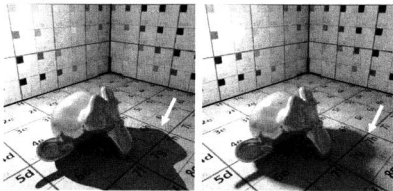


图 15-67 不同光源中阴影的边缘效果



Spot（聚光灯）也是一种常用的点光源，可以通过参数面板中的 Size（尺寸）和 Blend（混合）值，来调整光照的范围和阴影边缘。如图 15-68 所示，为其参数面板。



图 15-68 Spot（聚光灯）的属性面板

对于 Sun（太阳）光源，渲染时 Yaf(a)ray 将根据光线的前进距离来计算照射区域的阴影，距离越远的地方，阴影的边缘越模糊。如图 15-69 所示，左图为未使用太阳光照明的效果，而右图则为使用太阳光的照明效果，请注意观察图中离窗户最远处位置的阴影边缘差别。

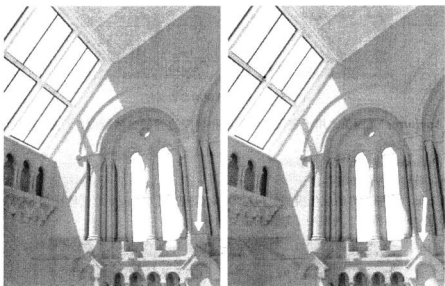


图 15-69 Sun（太阳光）产生的阴影效果

Area（面）光源是在室内和建筑表现中应用最广泛的一种光源，它能产生十分平滑的阴影边缘过渡，并且可以在物体表面制造十分清晰的反射高光边缘。同时，对比 Spot 和 Point 等光源的应用性能，面光源会花费更多的时间来执行渲染计算。最后，在调节面光源的时候，还需要注意：不要在场景中对光源做缩放操作，一定要使用参数面板中的 Size（尺寸）控制来调节。如图 15-70 所示，为 Area（面）光源的 Size 控制面板。

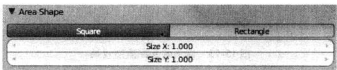


图 15-70 面光源的 Size（尺寸）控制面板

Area（面）光源与 Mesh Light（物体光源）的区别在于，前者将和其他灯光物体一样，不会在渲染中被渲染显示出来，而后者作为一个物体，可以直接被渲染出网格本体。

## 15.4.2 镜头参数

当选择场景中的镜头物体后，可在渲染面板中找到针对摄像机的特定选项。如图 15-71 所示，为 Camera（摄像机）的参数设置面板，在这里可以选择不同类型的镜头模式。



图 15-71 Camera（摄像机）参数的设置面板

### 15.4.2.1 Perspective（透视）模式与镜头 DOF（景深）

**Perspective:** 透视模式是标准的镜头模式，用于模拟真实世界中的镜头成像效果。如图 15-72 所示，为 Perspective（透视）模式中提供的 DOF（Depth of Field，景深）参数面板。

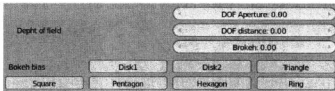


图 15-72 DOF 景深的参数面板

**DOF Aperture:** 光圈值，用于定义聚焦物体和模糊边界的过渡范围，一般来说值应控制在 0.5 以内。

**DOF distance:** 定义景深的计算距离。

**Bokeh:** 模糊边界的旋转角度。

**Bokeh bias:** 模糊边界的成像模式，包括 Disk（凹凸）、Triangle（三角）和 Square（直角）等模式。如图 15-73 所示，为使用了景深的渲染效果。

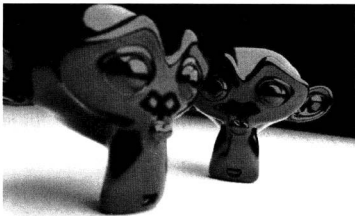


图 15-73 镜头的景深渲染效果

### 15.4.2.2 Orthographic（正交）模式

**Orthographic:** 正交模式下的镜头将取消透视效果，画面的效果将与视图中的正交模式相同。如图 15-74 所示，为 Orthogonal 的参数面板，其中 Scale（缩放）参数用于缩放镜头与物体间的距离。

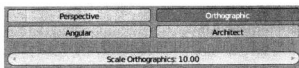


图 15-74 Orthographic (正交) 模式的参数面板

如图 15-75 左图所示, 为透视模式下的物体视图效果, 而右图为正交模式下的物体视图效果。

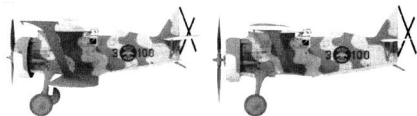


图 15-75 透视与正交模式的视图区别

#### 15.4.2.3 Angular (广角) 模式

**Angular:** 广角模式, 可以用于制作鱼眼镜头等可视范围在 180 度的特殊画面。如图 15-76 所示, 为 Angular 广角模式的参数面板。

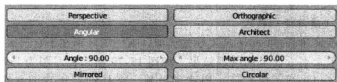


图 15-76 Angular (广角) 模式的参数面板

**Angle:** 角度, 用于控制镜头的水平角度, 系统将自动根据相机的宽高比来计算垂直角度。

**Max Angle:** 镜头的最大对角角度, 如图 15-77 所示, 为镜头的对角示意图, 其中斜对角线为镜头的最大对角角度。

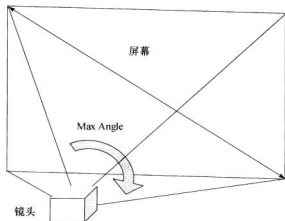


图 15-77 镜头的对角示意图

**Mirrored:** 是否创造 X 轴上的镜像。

**Circular:** 是否创建圆形的渲染边界。

如图 15-78 所示, 为艺术家 Gabich 利用广角镜头制作的特殊场景效果。



图 15-78 Angular (广角) 镜头的渲染效果

#### 15.4.2.4 Architect (建筑师) 模式

建筑师模式的效果和 Perspective 透视模式类似, 唯一的区别在于它忽视了垂直视角上的透视分量, 而只保留了水平方向上的透视效果。如图 15-79 所示, 左图为 Perspective (透视图) 效果, 而右图则为 Architect (建筑师) 模式的效果, 请仔细对比它们在垂直方向上的透视效果。

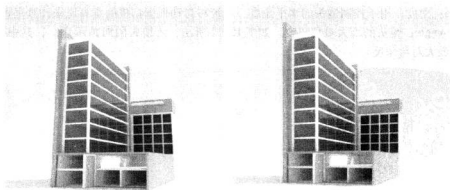


图 15-79 Perspective (透视图) 模式与 Architect (建筑师) 模式的效果区别

### 15.4.3 设计场景灯光与镜头

首先单击 Shift + A, 在场景中添加一个 Sun (太阳) 灯光, 使用快捷键 G 和 R 调整它的位置和角度, 让光线能直接射进窗户, 如图 15-80 所示。

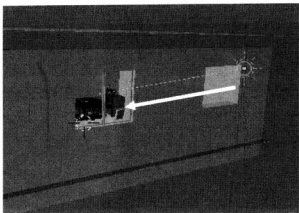


图 15-80 设置 Sun (太阳) 光的位置和照射方向

为 Sun 光选择一个淡蓝色的颜色, 用于模拟天光的色彩效果, 保持其他设置为默认, 具体参数如图 15-81 所示。

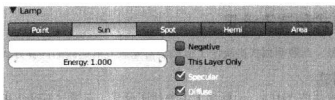


图 15-81 Sun (太阳) 光的参数设计

接着在两个窗户的位置分别添加一个面光源, 用做辅助光源, 在 Area Shape 面板中修改其 Size 参数来调整大小, 并使用 G 和 R 调节它们的照明方向为垂直于窗户指向室内, 效果如图 15-82 所示。

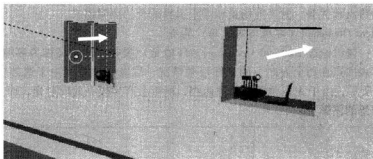


图 15-82 在窗户位置添加 Area 面光源

如图 15-83 所示, 设置面光源的照明属性, 保持大部分为默认即可, 并且开启阴影计算。

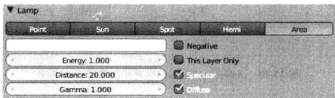


图 15-83 Area (面) 灯光的参数设置

在桌子的正上方再添加一个 Area（面）光源，作为室内的照明光源，如图 15-84 所示。

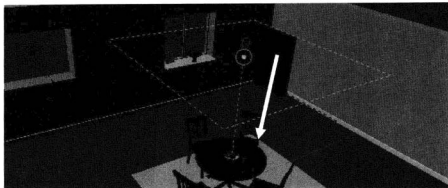


图 15-84 添加室内光源

最后，将镜头参数设置为标准的 Perspective（透视）模式，并启用 DOF（景深）效果，使用 Distance（距离）参数将焦距定位在桌子的位置。

## 15.5 照明模式和渲染设置

灯光决定了光源，而照明模式则决定了渲染器对光源发射光线的计算方式。

### 15.5.1 照明模式

使用光线追踪算法的 Yaf(a)ray 提供了多种照明模式，它通过 Global Illumination（全局照明）的形式，充分利用场景中有限的光线资源，来完成对整个场景的自然光模拟，还原物理光照的效果。GI 全局照明最大的优势，就是可以处理漫反射物体发射到其他物体上的非直接照明效果，同时还能在计算中避免相邻表面由于色彩扩散而产生的溢色现象。但是，目前 Yaf(a)ray 还不能提供一个完美的 GI 照明算法模型，所以需要深入了解每一种照明模式的原理，分别利用它们之间不同的特点和优势，为当前场景和需求选择适合的照明模式。

#### 15.5.1.1 Ambient Occlusion（环境闭塞修正）算法

在第 5 章中，曾讨论过 Ambient Occlusion（环境光）吸收，这里又称为环境闭塞修正算法，它常被用于计算表面点在每个方向上的光线反射情况，以及物体之间由于遮挡而产生的光能衰减。如图 15-85 所示，由于 A 点位置无物体遮挡，所以会产生较亮的反射光，而 B 点由于球体的遮挡将会产生较黑的阴影。

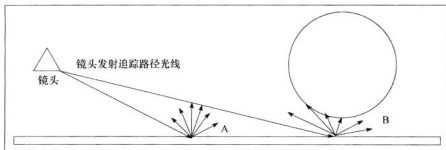


图 15-85 Ambient Occlusion（环境闭塞修正）算法

AO 可以模拟全局光照下的间接灯光照明，是一种十分快速的全局光照方式，可制作作为后期使用的明暗调节图等，同时也可用于计算焦散效果。如图 15-86 所示，为 AO 的参数面板。

**Samples:** 取样率, 设置用于计算闭塞物体的光线数。采样率越高, 渲染结果越精确, 但是需要消耗较多的计算时间。

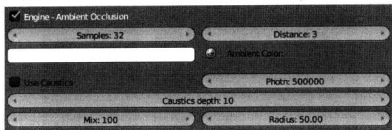


图 15-86 Ambient Occlusion 的参数面板

**Distance:** 设置用于计算的闭塞光线长度, 值越大则渲染器将花费更多的时间, 对环境内更大范围内的物体做采样计算。

**Ambient Color:** 用于选择 AO 的颜色。

**Use Caustics:** 开启焦散计算功能。

**Photon:** 用于计算焦散效果的光子数, 光子数越多, 计算量越大, 光子图的分布越均匀。Photon 光子运算是一种不同于路径追踪的光线处理过程, 如图 15-87 左图所示为光子的发生原理图, 光子是一种光能粒子, 它从光源发射出来, 在场景中做碰撞与反弹, 直到能量衰减至一定数值后而停止反弹计算。它的优势就在于它比路径追踪计算更容易形成焦散效果。如图 15-87 右图所示, 为光子映射图的原理图。光子映射图就是大量光子从光源发出后, 在输入光线方向上被撞击位置处形成的一个映射区域。不同于路径跟踪的无偏差计算, 这是一种有偏差的递归运算过程, 因为光子的反弹方向和能量衰减是非线性的, 但是如果能够实现持续光子的汇聚, 即可形成接近真实的物理光线效果。

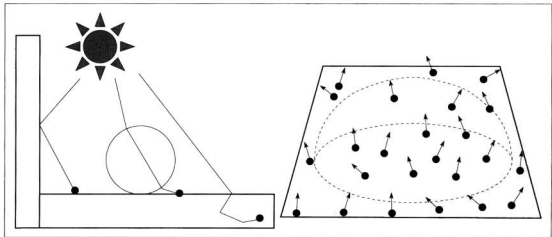


图 15-87 光子与光子映射图原理

**Caustics depth:** 焦散深度值, 用于定义光子的反射与折射量。

**Mix:** 用于混合的光子量, 值越大渲染的时间越长。

**Radius:** 光子图的混合半径, 值越低渲染精度和细节度就越高, 焦散光子的映射也会更平滑, 但是需要提高光子的数量, 才能使搜索系统可以利用更多的光子来参与混合计算。

如图 15-88 所示，为使用 AO 模式渲染的焦散效果。

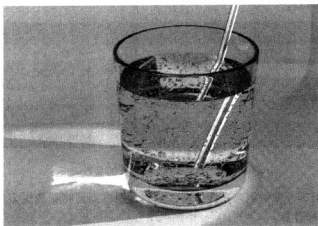


图 15-88 透明物体背后的焦散渲染效果

#### 15.5.1.2 Path Tracing（路径跟随）算法

Path Tracing（路径跟随）算法是一种无偏差的光线追踪算法，光线将从镜头开始，经过一级或二级反射后就能找到光源的位置。如图 15-89 所示，为 Path Tracing（路径跟随）在场景中计算光线的原理图，光线在每一个接触点都将随机发射二次反射光线，并进行采样计算，以区分光线是与其他表面进行了二次反射，还是直接查找到了光源位置。

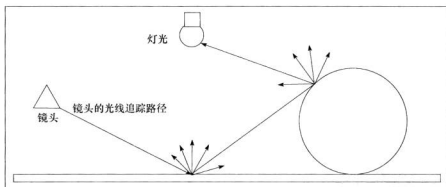


图 15-89 Path Tracing（路径跟随）算法

由于路径跟随的计算量小，所以更适合开放式的场景。在使用中应尽量采用大范围的区域光照明，避免使用聚光灯等点聚照明光源，这样才能增加路径光线寻找到光源的机会。例如当配合使用 HDRI 或日光照明时，画面看上去就会如同穹顶照明一样，清晰而明亮，而且渲染的运算时间也非常短。如图 15-90 所示，为 Pathtracing（路径跟随）模式的参数面板。

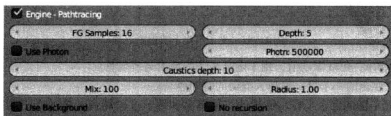


图 15-90 Pathtracing（路径跟随）模式的参数面板



**FG Sample:** 终光采样率, 由于 Path Tracing (路径跟随) 模型的光线是从镜头射出, 所以这里的采样率用于设置镜头的像素采样率。

**Depth:** 光线的反弹深度, 较大的反弹深度值可产生较多的反弹次数, 让光线在场景中使用更多的反弹路径去寻找光源, 令画面的暗处更亮, 一般将其设置为 3 至 4 即可。如图 15-91 左图所示, 为将光源放置在开阔位置处的照明采样效果, 可以看到, 2 次反弹深度的计算量足够照亮大部分场景, 每个角落都能在追踪采样下很快地查找到光源; 而右图则是将光源放置在非开阔位置处的采样效果, 由于光线的反弹次数较少, 画面部分区域没有足够的反弹光线去查找光源, 场景也将因此无法得到照明, 同时还会产生许多的噪点, 特别是离光源越远的地方, 噪点现象越严重。

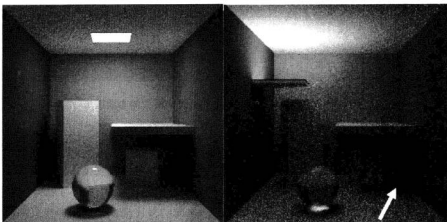


图 15-91 光线的 Depth (反弹深度) 对照明效果的影响

**Use Background:** 使用背景光源。

**No recursion:** 取消光线追踪的阴影递归运算。

#### 15.5.1.3 Photon Mapping (光子映射) 算法

Photon Mapping (光子映射) 算法是一种更高效的路径跟随算法, 因为它在 Final Gather (最终光线采集) 计算的辅助下, 不仅仅计算了从镜头中发射出的反向追踪光线, 还叠加了由光源处发射的光子映射。如图 15-92 所示, 为光子映射算法在场景中进行运算的原理图。

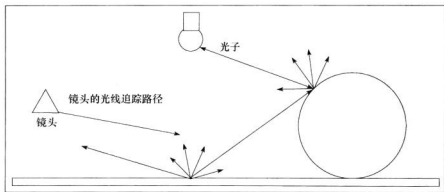


图 15-92 Photon Mapping (光子映射) 算法

全局的光子映射计算可以构建一个低分辨率的光线辐射映射图, 而终光采集则可以在这基

础上进行插值运算,使光子图的分布更加均匀。如果在一定区域上进行了光线优化的集中光子数目足够多,渲染时将有效地提高全局光在画面中的计算精度,因此这种算法十分适合封闭式的场景。如图 15-93 所示,为 Photon Mapping (光子映射) 模式的参数面板。

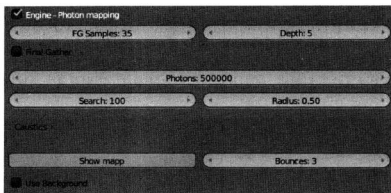


图 15-93 Photon Mapping (光子映射) 模式的参数面板

**FG Samples:** 终光采样率。

**Depth:** 反射与折射的深度值,这里要注意:较大的深度值会造成光子映射图曝光过度。如图 15-94 左图所示,为设置 Depth (深度) 值为 12 的光子图,可以看到由于深度值过大,场景中的每一个角落都得到了过量的采样运算,这种没有明暗对比的光子图反而对渲染没有任何实际意义。所以恰当的深度值可以为渲染提供更准确的间接照明对比度。如图 15-94 右图所示,是设置 Depth 为 5 的光子图。

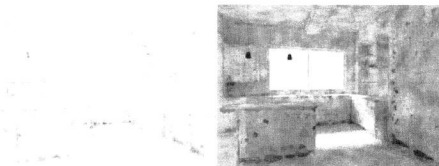


图 15-94 合理的光子 Depth (深度) 值可避免光子曝光过度

**Final Gather:** 终光采集,将对每个最终采集点的相邻光子开启平均化处理,从而产生一个均匀的光子映射插值图,用于辅助渲染计算。

**Photons:** 调节参与运算的光子数。

**Search:** 通过设置 Search 值可以调整参与运算的检测光子数,控制系统在寻找光子混合数时的插值搜索值,使相邻光子的色彩能够得以相关匹配,合适的取值范围为 50 至 150。如果匹配的相关性越大,色彩的过渡也越自然;反之,色彩的过渡反差就会越大。因此如果设置的搜索值过低,则色彩间的匹配相关性也会相应降低。如图 15-95 左图所示,为搜索值为 20 时的光子图,而右图则是将搜索值提高至 150 时的光子效果图。可以看到,光子由于提高了搜索值而获得了较高的匹配相关性,画面的色彩明显比左图看上去更平滑和自然。

**Radius:** 为了进一步控制光子的匹配值, 这里还可以使用 Radius 来定义运算时采样的漫射半径范围。如果半径足够小, 那么光子的映射将更精确。但是如果此时的搜索值设置偏低, 光子则会在平滑计算后表现出低频的噪点, 原因在于当前搜索采样区域内没有足够数量的光子数, 如图 15-96 左图所示。解决的办法是提高光子数, 或者增大搜索值, 效果如图 15-96 右图所示。

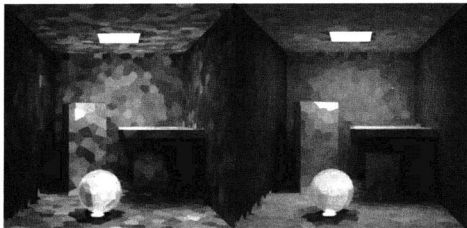


图 15-95 Search (搜索) 值对光子图的匹配相关性影响



图 15-96 调节 Radius (漫射半径) 来获得平滑的光子图

Search (搜索) 值和 Radius (漫射半径) 值是两个影响光子图效果和运算性能的重要参数, 需要在光子预览中根据色彩匹配性的检测结果, 来寻找到两者之间的平衡调节点。较小的匹配相关性是由于受到了 Radius (漫射半径) 的限制, 如图 15-97 左图所示房间的角落黑色区域, 因为这个区域的漫射半径内没有足够数量的光子能参与运算。而过大的匹配相关性则是受到 Search (搜索) 值的影响, 如窗口位置曝光过度的区域, 因为这里的搜索值使用了过量的光子来参与平滑运算。一幅匹配度合适的光子渲染检测图, 可以看做像是印象派风格的油画, 其中相邻区域的亮度和对比度都十分匹配近似, 并且无黑点区域, 如图 15-97 右图所示。

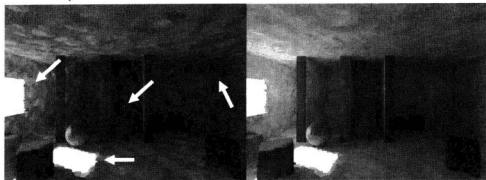


图 15-97 光子图的匹配相关性检测

如果增加光子的发射数，一定能提高光子映射的质量，但是这会增加更多的渲染时间。当 Radius（漫射半径）越小时，运算的精度越高，效果也越好，但是如果没有足够多的光子去计算匹配相关性，就会产生大量的噪点。所以这里既需要增加光子数量来弥补效果缺陷，又同时要适当地增加搜索值，用来平滑检测时由于漫射半径过小而造成的噪点效果。如图 15-98 所示为一个符合追踪渲染要求的光子效果图，其中光子数为 2 000 000，Radius（半径）为 0.1，Search（搜索）值为 150。

**Show mapp:** 开启显示光子图的分布。

**Bounces:** 用于定义缓冲光子时的反弹次数，一般设置为 3 或 4。

**Use Background:** 使用背景照明，这是一种在 GI 照明中使用间接照明的方式，如果是露天场景，则一定要开启背景照明模式。

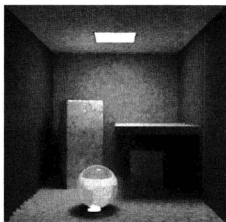


图 15-98 优化光子的计算流程来提高光子的匹配相关性

## 15.5.2 照明的常规参数

在渲染面板中，可以找到渲染的常规选项参数面板，如图 15-99 所示。

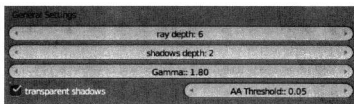


图 15-99 常规渲染的参数面板

**Ray Depth:** 光线追踪深度，如图 15-100 所示，为光线路径追踪运算时的深度原理图。当设置深度值为 1 时，光线只会被计算 1 次反射或折射效果，那么按照物理原理，光线本应经过的点 2 位置将不会被计算至光线追踪路径中，从而导致渲染结果产生误差，通常这个参数会影响物体材质的 Mirror（反射）和 Transparency（透明）属性。

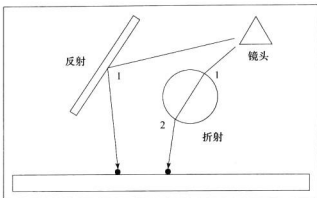


图 15-100 Ray Depth (光线追踪深度) 的计算原理

**Gamma:** 伽玛值用于校正渲染输出的色彩标准，完成色彩的线性工作流。在 Windows 系统下常设置为 2.2，而在 MacOS 或 Linux 系统下则需要设置为 1.8。

**Transparent Shadows:** 透明阴影是由物体在 IOR (折射率) 影响下产生的虚假阴影效果，它需要配合 Shadow Depth (阴影深度) 参数来使用，用于显示物体表面在全局光照算法下产生的光线过滤和焦散效果。在 Yaf(a)ray 中，可以使用两种方法来制作这样的假阴影效果，第一种是在 Shiny 模式下增大 Transparency (透明) 值，第二种是在 Glass 模式下开启 Fake Glass Shadow (伪阴影) 选项。

Shadow Depth: 阴影深度，用于控制光线穿过物体表面产生的透明假阴影深度。

#### 15.5.2.1 Transparent Shadow (透明阴影)

接下来，讲解如何通过不同设置，来制作透明物体的阴影效果。

首先采用 Area (面) 光源照明，如图 15-101 左图所示，为在不开启 Ray Tracing Shadow (光线追踪阴影) 和 Caustics (焦散) 计算时的玻璃阴影渲染效果。可以看到虽然反射光能被正常渲染出来，但是阴影的折射光却没有显示出来。如图 15-101 右图所示的粗体线段，为这种方式下光线的计算路径图，由于没有开启光线追踪阴影选项，渲染将无法计算阴影至光源的光线路径。

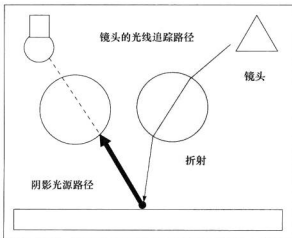
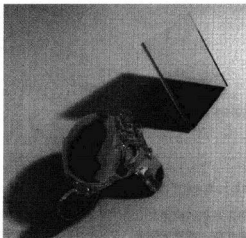


图 15-101 缺少折射光计算的阴影效果

接着，开启模式中的 Caustic (焦散) 计算，如图 15-102 左图所示，由于玻璃材质的 IOR 为

1.55, 大圣物体又是近似球体形状, 光子在它的后面就形成了聚焦, 但是旁边的玻璃物体却在阴影位置产生了不均匀密度的光子分布。这是因为光子是由灯光直接发射产生的, 如图 15-102 右图所示, 虚线线段为光子的前进路线, 当光子没有得到聚焦等均匀化处理时, 就会直接发散性地叠加在渲染的阴影结果中。

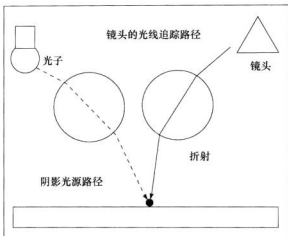
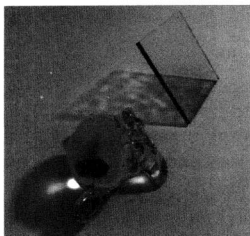


图 15-102 启用光子追踪的 Caustic (焦散) 效果

为了解决均匀体积物体的阴影问题, 可以考虑舍弃使用光子参与的焦散运算, 而是开启材质的 Fake Glass Shadow (伪阴影), 并设置足够大的 Shadows Depth (阴影深度)。这样的效果如图 15-103 左图所示, 透视阴影的分布十分均匀, 但是看上去却并不真实。如图 15-103 右图所示, 可以看到, 这种光线阴影其实并没有按照物理原理计算, 而只是形成了一个“伪造”的假阴影。

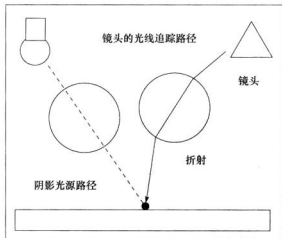
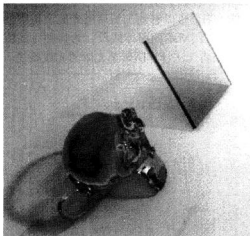


图 15-103 使用 Fake Glass Shadow (伪阴影) 来实现平滑的阴影效果

如果既要实现光子焦散, 又需要一个均匀平滑的阴影效果, 应该怎么办呢? 换个角度来观察当前场景, 对于光源部分, 由于之前选择的一直都是面光源, 或者太阳光, 它们的特点就是光子的发射方向是向四面八方散射的。因此可以考虑将光源更换为 Spot (聚光灯), 由于它的聚光特性, 可以将光子在发射方向处集中起来。如图 15-104 左图所示, 当关闭了 Glass Fake Shadow (伪阴影) 和 Transparent Shadow (透明阴影) 选项, 改用聚光灯照明时, 就得到了一个十分真实的

玻璃阴影运算效果。如图 15-104 右图所示,其中虚线线段为这种设置下光子的前进路线,完全满足物理模拟的原则。

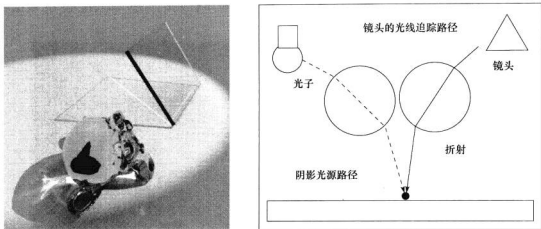


图 15-104 改变光源属性来获得真实的玻璃渲染效果

通过不同的设置,可以获取效果各异的阴影,读者在实际的应用中,应比较并理解这几种方式在性能和效果上的优劣,根据自己的情况选择适合的方式来制作阴影效果。

#### 15.5.2.2 Anti-aliasing (抗锯齿)

Anti-aliasing (抗锯齿),是以重构连续图像为目标的一个采样计算过程,它可以减少画面中物体边缘的锯齿感,解决物体的细节失真。Yaf(a) ray 基于 Low-discrepancy Sampling (低差异采样)的算法,将画面中的锯齿转换为噪点,而人眼对于噪点缺少敏感性,渲染因此完成了对画面的抗锯齿处理。

AA Threshold (Anti-aliasing Threshold, 反锯齿的运算阈值),它是基于相邻像素的差异性,用来控制抗锯齿运算时,色彩像素值间的取样插值。锯齿效果本身就是在 3D 场景中,由于非连续采样而导致的逼近性误差。而渲染又是以像素为一个基本处理单元,所以抗锯齿计算就是当像素间的色彩亮度差大于设置的极限阈值时,系统将对这部分的像素点做重采样计算,直到运算后的差值符合预设的阈值标准,这一过程又被称为自适应采样抗锯齿计算。如图 15-105 左图所示,为渲染时由于像素点过大,每个像素单元中的采样计算无法完成色彩修正,造成边缘锯齿效果的原理图,其中每一个方块代表当前画面中的像素点尺寸,方块中的点代表了采样数。如图 15-105 右图所示,则为这种低采样计算造成的锯齿效果。

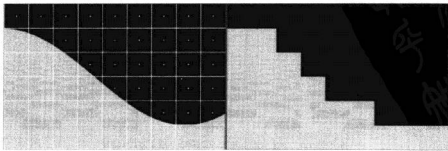


图 15-105 像素锯齿的形成原理

如果降低 AA Threshold (阈值),渲染时系统将消耗更多的时间和内存来执行重采样计算。

如图 15-106 左图所示, 为设置 AA Threshold (阈值) 为 0.05 时的渲染效果, 其中白色区域为被计入重采样的像素范围。如图 15-106 右图所示, 为降低设置为 0.01 时的渲染效果, 可以看到, 由于阈值减小, 需要被重采样的边缘区域增大, 有些过渡不够平滑的阴影和间接光照区域也被计入重采样的范围。

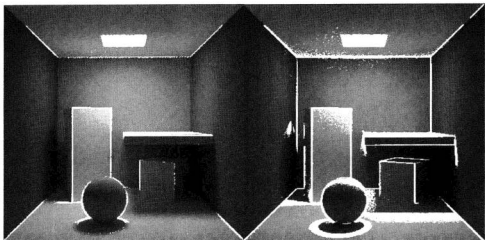


图 15-106 AA Threshold (反锯齿运算阈值) 效果

除了阈值控制, Yaf(a)ray 还提供了抗锯齿过滤选项, 用于二次平滑物体的边缘效果。如图 15-107 所示, 为 Anti-aliasing (抗锯齿) 的参数面板。

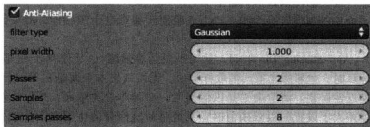


图 15-107 Anti-aliasing 抗锯齿的参数面板

**Filter type:** 过滤器类型, 用于辅助 AA 计算时对像素点进行取样和筛选。常用的过滤器为 Box (矩形)、Gaussian (高斯) 和 Mitchell-Netravali (米歇尔) 型, 如图 15-108 所示, 分别为使用这三种过滤器的抗锯齿效果。

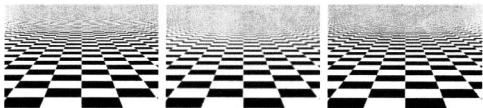


图 15-108 不同过滤器对边缘的抗锯齿处理效果

**Box:** 矩形滤波会同等重视所有的像素样本, 而且运算速度快, 但是在处理某些特殊类型的噪点时效率不高, 反而容易产生 Post-Aliasing (后期锯齿)。



**Gauss:** 高斯滤波可以提供一个较光滑的采样结果,但是通常会导致图像出现轻微的模糊,这些模糊在一定程度上也可以帮助掩盖剩余的图像锯齿。

**Mitchell-Netravali:** 米歇尔滤波能产生三者中最好的采样效果,并且可以提高边缘的清晰度,推荐在最终渲染中使用。

**Pixel width:** 采样滤波器的有效值,也就是滤波器横轴上参与有效过滤的波峰宽度值,常用于控制采样时的滤波性能,如图 15-109 所示。较窄的像素宽度值会降低单位面积上的有效采样率,造成边缘得不到足够的采样计算而变得尖锐,而增大像素宽度值可以增大单位面积上的采样率,让画面边缘显得更平滑,通常设置值为 1.5 至 1.9 之间。

较低宽度值的滤波器,会需要更多的渲染时间进行采样,以完成对图像的重构。如图 15-110 左图所示,是使用值为 2.5 时的采样边缘,渲染时间为 49 s;而中图是值为 1.5 时的效果,需要 54 s 的渲染时间;右图是设置值是 1.1 的效果,它的边缘最尖锐,渲染时间也更长为 82 s。

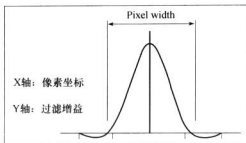


图 15-109 采样滤波器的波形图

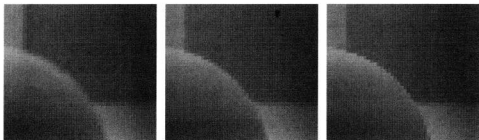


图 15-110 Pixel Width (像素宽度) 对边缘的计算性能

**Passes:** 执行抗锯齿运算的通道数。

**Samples:** 采样数,值越大,光线追踪的二级光线路径中将产生越多的交叉点和取样次数。如图 15-111 左图所示,为使用 Samples 值是 1 的渲染效果,右图是 Samples 设置为 32 的效果。请注意两图中阴影和球体的边缘部分,可以看到,采样率的增大明显减少了画面的锯齿。

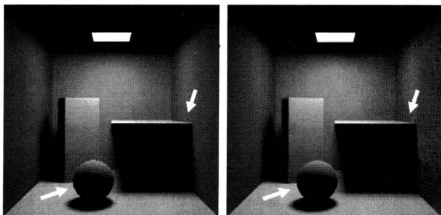


图 15-111 提升 Samples (采样数) 来减少边缘的锯齿

**Samples passes:** 用于执行采样运算的通道数。

### 15.5.3 场景渲染设置

接下来就根据场景来选择合适的渲染参数，如图 15-112 所示，为抗锯齿的参数面板。



图 15-112 AA（抗锯齿）的参数面板

常规选项中，设置参数如图 15-113 所示。

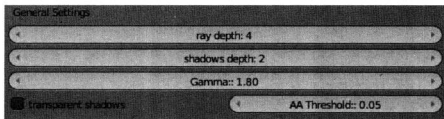


图 15-113 常规渲染选项的参数面板

最终渲染使用光子映射模式，如图 15-114 所示，为光子映射的参数面板。

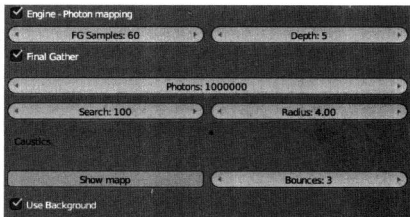


图 15-114 光子映射的参数面板

最后执行 Render 按钮即可开启渲染计算，当前的设置可以得到一个正常的渲染效果。为了方便合成，再次修改光照模式为 AO，渲染一个明暗对比图。如图 15-115 所示，为 AO 渲染的参数设置面板。



图 15-115 AO 渲染设置的参数面板

两种渲染结果如图 15-116 所示。

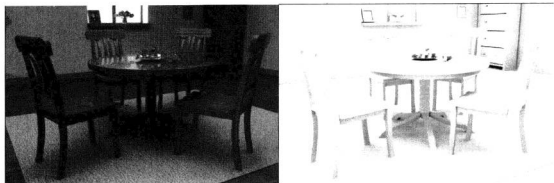


图 15-116 两种模式的渲染输出

## 15.6 后期处理

切换到结点编辑器中，使用图片输入结点将两个渲染结果输入至结点系统，如图 15-117 所示。添加一个 Multiply（叠加）结点，分别连接两个渲染图层，并调节合适的 Fac 值将 AO 明暗对比图叠加至原图上，用于强化画面的亮度和色彩对比效果。

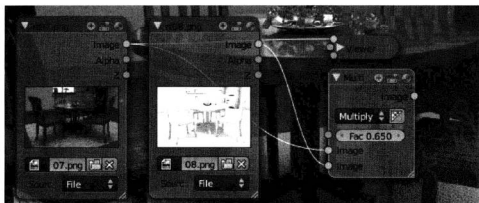


图 15-117 使用图片输入结点将渲染结果输入至结点编辑器

接着输出一个混合后的结果至 Hue 结点，调节画面的对比度，并使用 Blur 结点将其做模糊处理，最后再使用 Overlay 结点将其与原输出叠加起来，用于调整画面的色彩。如图 15-118 所示，背景的左半部分为调节后的效果，而右半部分为叠加调节前的效果，可以看到，经过处理的画面色彩更鲜艳自然。



图 15-118 通过叠加模糊图层来调节画面的色彩

最后，通过输出结点，就可以得到如图 15-119 所示的最终效果图了。



图 15-119 最终室内效果图

## 15.7 本章小结

本章着重讲解了在 Blender 中，使用 Yaf(a)ray 外置渲染器制作室内等场景效果图时，如何

对材质和渲染等参数进行选择 and 配置。通过对每一个功能原理的分析和讲解，希望读者能深入领会每个参数在渲染中的意义，而不是仅仅去记忆参数数字，从而掌握如何使用和调节。读者在完成本章学习后，需要熟练掌握的基本技能包括：

- ❑ 建模中，掌握 Array（阵列），Mirror（镜像）和 Solidify（实体化）修改器的应用。
- ❑ 理解 4 种材质着色器，熟悉各种类型的材质适合选用的着色器。
- ❑ 利用模型适当地弥补无法由材质单独完成的物理模拟，例如液体与玻璃接触位置的水面张力细节。
- ❑ 掌握贴图的混合技巧，理解纹理在通道上的应用。
- ❑ 理解不同灯光的阴影效果，熟悉灯光的基本参数调节。
- ❑ 理解不同的照明模式及相关参数概念，包括光子、焦散、光线追踪、抗锯齿等，能结合对参数的理解完成对渲染结果的性能优化。

## 第 16 章

# 角色动画

本章将介绍如何使用 Blender 制作简单的角色人物动画。我们将在一个低模角色的基础上，讲解如何设计可控骨骼，如何与人物模型完成绑定操作，最后编辑骨骼动作用于制作步行动画。在骨骼设计中，将重点讲解一些常用的约束控制，例如 IK 反向运动约束设计等；在动画制作中，将简介如何使用 Dope Sheet（动作编辑器）和 NLA Editor（非线性编辑器）工具来制作一个基本的键帧步行动画。

## 16.1 角色模型制作

首先制作角色模型。

### 16.1.1 头部与身体建模

制作动画模型时一定要注意模型的重心位置，因为当完成网格绑定后，骨骼将以网格的重心为控制基点。如果两者的重心不在同一个位置上，在骨骼与网格之间就可能出现较大的运动偏移现象。所以在开始制作之前，首先需要单击 Shift + C 将光标重置到坐标轴的原点处，然后再按 Shift + A 添加一个 Cube 物体。接着在修改器面板下为其添加一个 Mirror（镜像）修改器，使用 Ctrl + R 对 Cube 做细分操作，如图 16-1 左图所示。完成切割后，使用快捷键 G 调节点的位置，制作出头部的形状，最后的效果如图 16-1 右图所示。

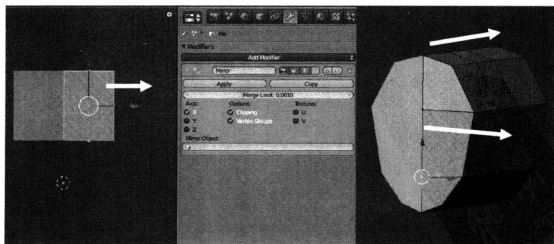


图 16-1 制作头部

为其添加一个 Subsurf（细分）修改器，修改网格的结构，大致效果如图 16-2 左图所示。接

单击修改器上的小眼睛，隐藏修改器的作用效果，然后使用鼠标 MMB 转向头部的底部视角，选择底面并按 E 向下做挤压操作，制作出脖子，如图 16-2 右图所示。

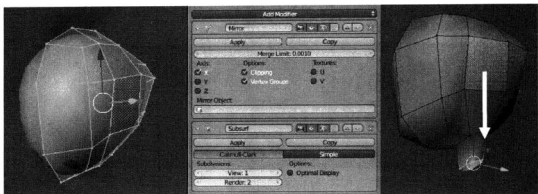


图 16-2 制作脖子

选择脖子底部的边，继续向下做一次挤压操作，然后分别对前面、侧面和后面三个方向的边向下做挤压细分，制作出胸部和背部部分，如图 16-3 左图所示。接着将前后两个面的边向侧方向做挤压，并在手臂处位置按 F 缝合起来，保留出一个 Loop 接口边，如图 16-3 中图所示。选择胸口底部的边，单击 E 继续向下挤压，制作出腹部和躯干部分，如图 16-3 右图所示。

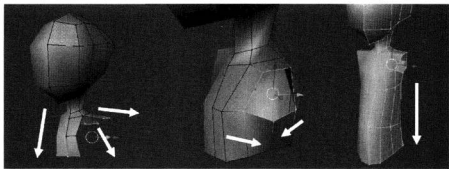


图 16-3 制作身体躯干

### 16.1.2 手臂建模

选择身体侧面上手臂的 Loop 结构边，单击快捷键 E + X 向 X 正轴方向挤压出手臂网格，如图 16-4 左图所示。接着在手臂上使用 Ctrl + R 做环形切割，配合快捷键 S 制作出上臂和手臂的粗细，以及肘部关节部分，效果如图 16-4 右图所示。

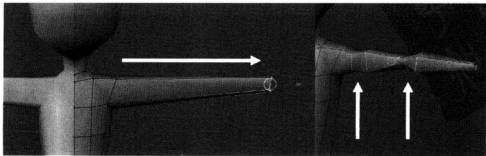


图 16-4 制作手臂

选取手臂上的前后两个对称边，单击 E + S 同时向外做挤压操作，制作出手掌的轮廓外形，如图 16-5 左图所示。接着使用 F 来缝合手背和手心里的面，用 Ctrl + R 对手掌做细分操作，如图 16-5 中图所示。最后缝合上下的空隙面，并对手部做切分和调整，然后选择侧面上的面，单击 E 向外挤压出大拇指的形状，最后的效果如图 16-5 右图所示。

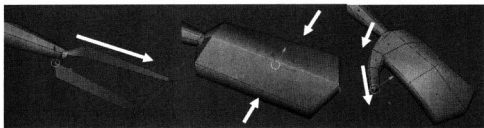


图 16-5 制作手掌

### 16.1.3 腿部建模

选择腹部前面的一个边，单击 E 向后挤压，并将其与背部的面缝合起来，制作出胯部，如图 16-6 左图所示。接着选取侧面的 Loop 边向外挤压，调整出大腿根部的面结构。由于这里的运动幅度较大，所以可以多添加几条 Loop 切割线，如图 16-6 中图所示。最后选取大腿根部最下面的边，单击快捷键 E 继续向下挤压出大腿和小腿部分，由于腿部的结构简单，不需要太多的布线，所以只需要简单地在大腿和小腿处缩放出粗细即可，最后的效果如图 16-6 右图所示。

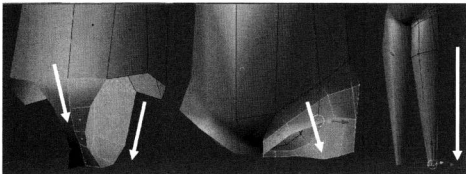


图 16-6 制作臀部和腿部

继续向下挤压小腿，然后选择小腿前面的边，向前挤压出脚背部分。接着选取小腿后部的边，向下挤压出脚跟，如图 16-7 左图所示。再选取前脚掌的环边向下挤压，如图 16-7 右图所示，最后在接口处单击 F 做面的缝合操作。

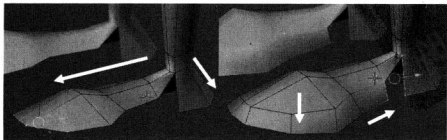


图 16-7 制作脚掌



由于在动画中,脚掌位置会出现弯曲等较大幅度的动作,所以可以使用 Alt + RMB 选取最下面的 Loop 边,按 E + G 向下再做一次挤压,然后单击 S + Z + Num0 将所有的点缩放至同一水平面上,再使用 Ctrl + G 将面移动到 X-Y 轴水平面上,如图 16-8 左图所示。最后使用 Ctrl + R 在膝盖处多添加几个 Loop 切割线,因为这里的关节运动需要较多的线条才能满足腿部弯曲时的网格正常变形。同时在大腿和小腿上再分别添加 Loop 切割线,单击 S 缩放出大腿和小腿的肌肉粗细,Loop 线排列和模型效果如图 16-8 右图所示。

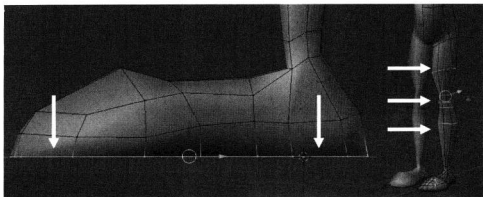


图 16-8 完成脚底板和膝盖部分

单击 Tab 退出物体的编辑模式,激活模型 Subsurf (细分) 修改器上的预览功能,最后的效果如图 16-9 左图所示。右图为细分后的线框效果,请注意观察人物运动关节处的布线密度。

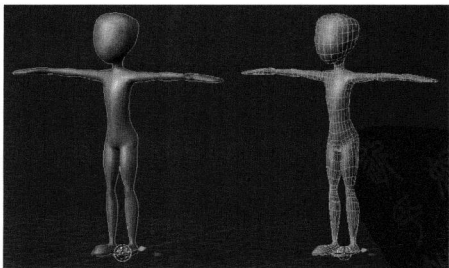


图 16-9 模型最终效果和线框图

低模人物模型到这里就做好了,读者也可以根据自己的需求为其添加更多的细节。

## 16.2 骨骼设计

本节介绍如何进行骨骼设计。

## 16.2.1 制作骨骼框架

接下来要为人物添加一套骨骼系统，用于控制人物的动作和姿态。

首先单击 Shift + C 将光标定位至原点处，使其与模型的原点处于同一个位置，然后单击快捷键 Shift + A 进入 Armature（骨骼）菜单，选择并添加一个 Single Bone 单骨头物体，如图 16-10 所示。



图 16-10 添加骨头物体

为了在编辑时能够更方便地查看骨骼，切换至骨骼的 Object（物体）面板，激活 X Ray 透视查看模式，如图 16-11 所示，这样角色网格模型将不会与骨骼产生遮挡效果。

### 16.2.1.1 制作躯干骨骼

单击 Tab 进入骨骼的 Edit Mode（编辑模式），按小键盘的 Num3 进入侧视图视角，使用鼠标 RMB 单击选择骨骼，如图 16-12 左图所示。接着单击 G 将骨骼整体移动至身体的躯干处，把骨骼的尾部移动至人物的髋部位置，把骨骼的头部移动至人物的头顶位置，如图 16-12 右图所示。

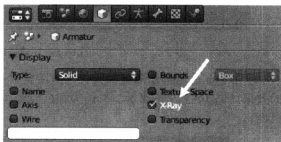


图 16-11 激活 X-Ray 透视查看模式

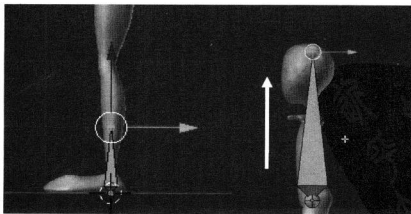


图 16-12 移动第一根骨骼的位置

单击 RMB 选择骨骼整体，使用快捷键 W 进入 Subdivide（细分）菜单，对骨头做细分操作，如图 16-13 左图所示。对于角色的躯干骨头数量，可以根据不同的需求做适当的调整。在本例中的骨头属于简化角色人物，所以可以使用一种常规的骨头设置——将头部设置为 1 根骨头，颈部 1 根骨头，躯干 3 根骨头。经过细分后的骨头调整设定如图 16-13 右图所示。

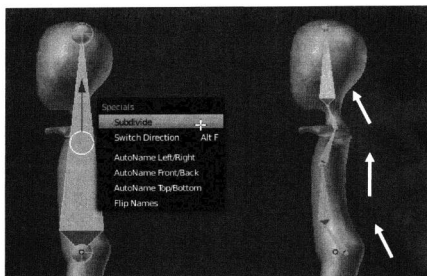


图 16-13 设置躯干和头部的骨头数量

### 16.2.1.2 制作手臂骨头

接着将光标移动至左肩位置，单击 Shift + A 添加一个新的 Bone（骨头），如图 16-14 左图所示。和身体躯干的细分制作方式类似，首先将骨头的头部移动至指尖位置，然后再使用细分工具，将手臂划分为肩部、上臂、下臂、手掌和手指等 5 个骨头部分，如图 16-14 右图所示。

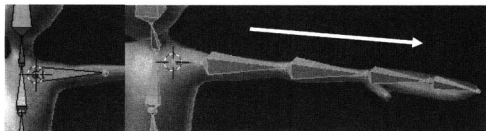


图 16-14 手臂的骨头设置

接着将光标移动至手腕位置，如图 16-15 左图所示，单击 Shift + A 添加一个新骨头，作为大拇指的骨头，最后的效果如图 16-15 右图所示。

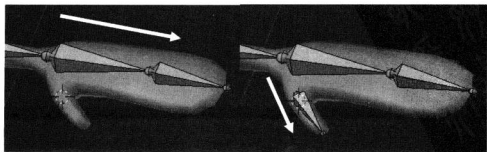


图 16-15 添加手指的骨头

## 16.2.1.3 制作腿部骨头

切换至正视图视角，将光标移动至髋部位置，如图 16-16 左图所示。但是这时的光标在 Y 轴上可能不在你所期望的位置，所以可以通过旋转至侧视图视角来定位光标，如图 16-16 中图所示。单击 Shift + A 在此处添加一个新的 Bone（骨头），移动骨头的头部至脚部位置，然后使用细分工具将骨头划分为大腿和小腿两个部分，如图 16-16 右图所示。

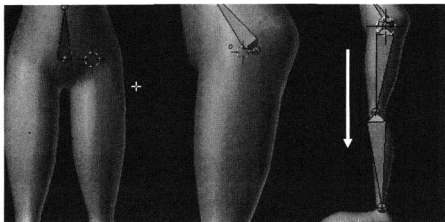


图 16-16 制作腿部的骨头

在侧视图视角中，将光标移动至脚掌上方的位置，如图 16-17 左图所示。单击 Shift + A 添加一个 Bone（骨头），并将其头部移动至小腿的根部位置，如图 16-17 右图所示。

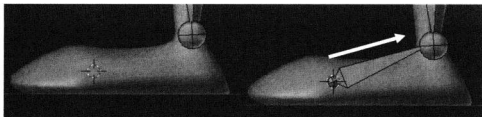


图 16-17 制作脚掌的骨头

保持光标的位置不变，再次单击 Shift + A，添加一个新的 Bone（骨头），移动骨头的头部至脚尖位置，如图 16-18 左图所示。接着选择小腿骨头的头部，单击 Shift + S 激活 Snap（吸附）菜单，选择 Cursor to Selected（将光标定位至选择的物体上），如图 16-18 右图所示。

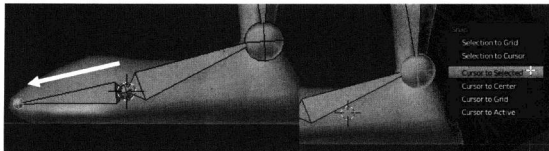


图 16-18 制作脚趾部位的骨头

在光标位置处再添加两个骨头，将骨头的头方向分别指向脚后跟和脚底，如图 16-19 所示为脚部骨头的最终效果。

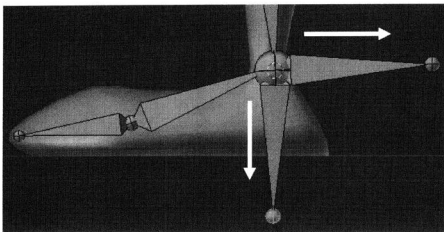


图 16-19 制作脚部骨头

## 16.2.2 制作骨骼约束

接下来开始为骨骼设计约束，首先在工具面板中进入 Armature（骨骼）属性，将 Display（显示）模式调整为 B Bone 模式，如图 16-20 所示。

### 16.2.2.1 骨头命名

接着在 3D 视图中，单击 N 调出骨骼的属性菜单，选择每个骨头，分别为其命名，如图 16-21 所示。

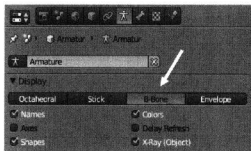


图 16-20 调整骨骼的显示模式

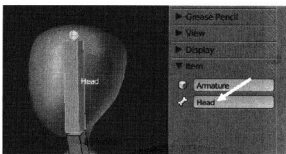


图 16-21 分别对骨头命名

骨头的命名十分重要，因为这影响在绑定模型和骨骼时，顶点组与权重值的命名设置与管理。一个良好规范的命名习惯，可以有助于实现网格与骨骼之间的快速协调和配置。如图 16-22 所示为对人物上肢、躯干和手臂的命名结果。其中对于手臂等具有左右对称关系的骨头，应在其名称后添加后缀“L”或者“R”，用于区分左右方向。

由于 B-Bone 的形状和尺寸比较统一，不方便区分，可以在单击 RMB 选择每个骨头的时候，使用快捷键 Ctrl + Alt + S 对骨头做缩放操作。例如将头部和胸部的骨头放大，而手指和脊椎等位置的骨头可适当缩小。具体的调整可根据个人的习惯来定，调整后的效果如图 16-23 所示。



图 16-22 为骨头命名

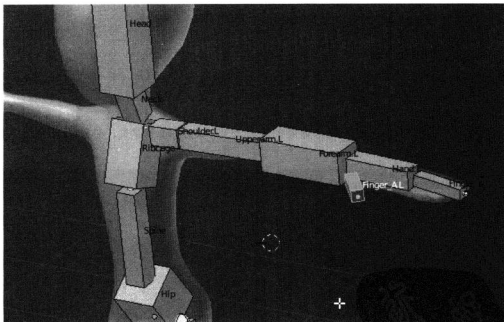


图 16-23 调整 B-Bone 的形状

#### 16.2.2.2 Segment (分段细分)

在颈部和脊椎的位置，只分别设置了一根骨头，但是参照现实中的生物体，这两个部位都由许多骨节组成。如果要使用多根骨头来制作这两个部分，无疑将会增加绑定后顶点组的数量，以及对人物动作的控制器数量。为了解决这个问题，转入骨骼的 Pose Mode（姿态模式），首先选择 Neck（颈部）骨头，如图 16-24 左图所示，在 Bone（骨头）面板中，激活 Deform（形变）属性，将 Segments（分段细分）的数值增大，此时颈椎骨头将会自动按照骨头方向，被划分为一个多个骨节的骨头物体，最后的效果如图 16-24 右图所示。

除了颈部和脊椎，也可以对下臂和手指的骨头分别做分段操作，使用一根骨头来模拟多骨节的结构，如图 16-25 所示。

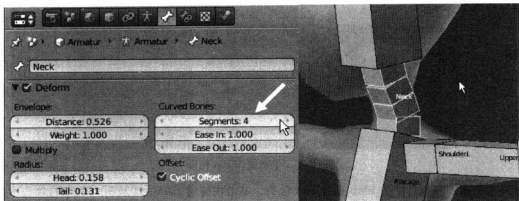


图 16-24 使用 Segments 对骨头做骨节细分

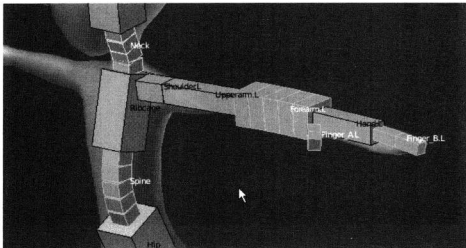


图 16-25 对活动关节做骨节细分

需要注意的是: Segments (分段细分) 对骨头的细分操作是处于权重级别的, 而并不会对骨头做物理上的细分。因此使用 Segments 不仅可以替代多骨头的功能, 而且不会增加骨头的数量, 减少了人物身上的控制器数量。网格上的权重分布也将按照 Segments (分段细分) 的细分数量, 做等比例的绘制, 因此在旋转骨头时, 顶点组上的点会产生平滑的旋转过渡。如图 16-26 所示为手掌的转动带动下臂上的骨节做递进的比例旋转。

#### 16.2.2.3 制作颈部约束

接下来制作 Hip (髋部) 骨头。在 Pose 模式下, 如果旋转 Hip 骨头, 它将以底部为轴, 带动整个上半身的躯干骨头做整体旋转, 如图 16-27 左图箭头所示。这是因为 Hip 的原点处于骨头的底部位置。这种运动效果并不符合生物体的结构特性, 我们希望 Hip 能在它的骨头顶部位置, 也就是以脊椎的最下方为原点做旋转运动。为了方便制作, 在骨骼的物体菜单中, 找到 Display (显示) 属性, 将显示模式修改为 Wire (线框)

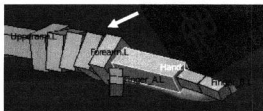


图 16-26 Segments 细分骨节的旋转效果

模式，如图 16-27 右图所示。

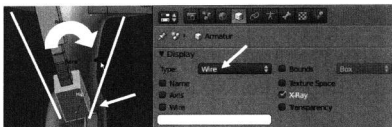


图 16-27 准备制作 Hip 骨头

转至侧视图视角，使用 RMB 选择 Hip 骨头，单击快捷键 Shift + D 复制一个新的骨头出来，然后松开鼠标不做任何移动操作，效果如图 16-28 左图所示。接着使用快捷键 Ctrl + Alt + S 将复制出来的新骨头缩小一定比例，并命名为 Hip\_Controller，如图 16-28 右图所示。

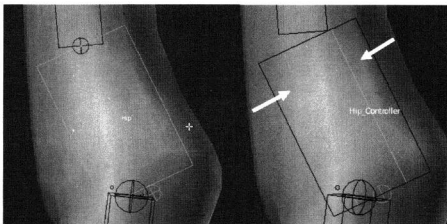


图 16-28 复制 Hip 骨头

接着选择 Hip 骨头，单击 W 选择 Switch Direction（骨头反向）功能，或者使用快捷键 Alt + F，如图 16-29 左图所示。反向旋转后的 Hip 将把物体的旋转原点转移至顶部位置，如图 16-29 右图所示，但是此时 Hip 并不能带动上体骨骼运动。

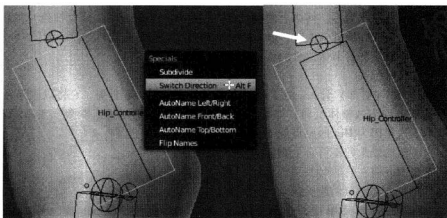


图 16-29 反转 Hip 骨头



为了控制 Hip 骨头的转动,就需要发挥 Hip\_Controller 骨头的功能。由于 Hip\_Controller 继承了原 Hip 物体的关联关系,所以它也具有控制整个上体躯干的功能。首先选择 Spine 骨头,如图 16-30 左图所示。然后单击 Shift + RMB 选择 Hip\_Controller 骨头,按 Ctrl + P 将两者 Connected (联接)起来,如图 16-30 右图所示。

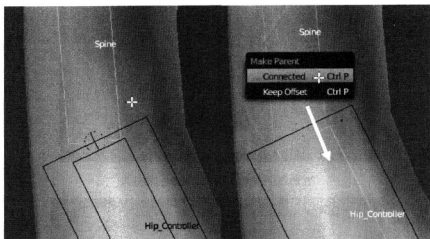


图 16-30 将 Spine 与 Hip\_Controller 建立父子关系

接着单击选择 Hip\_Controller 骨头,如图 16-31 左图所示,按 Shift + RMB 选择 Hip 骨头,使用 Ctrl + P 为其制作父子关系,并选择 Keep Offset (保持偏移量),如图 16-31 右图所示。

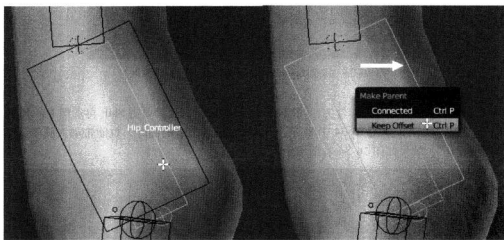


图 16-31 为 Hip\_Controller 骨头设置父子关系

那么这时再回到 Pose 模式下,选择并转动 Hip 骨头。由于子骨头 Hip\_Controller 具有对上肢骨骼的驱动作用,所以躯干将以 Hip 骨头的头部位置为原点做旋转运动,如图 16-32 箭头所示。

但是在转动 Hip 骨头时,Spine 和 Ribcage 骨头将由于联动性,被动地参与旋转运动,无法实现上半身与腰部的分离运动。为了消除这个关联问题,可以转至 Spine 的骨头面板,如图 16-33 左图所示,取消其 Inherit Rotation (旋转关联)的功能选项。这时再次选择 Hip 骨头并对其做旋转操作时,就会发现骨头的运动将不再带动 Spine 骨头做关联旋转运动了,如图 16-33 右图所示。而这样的效果更符合真实的骨骼运动,可以用于制作出臀部的摆动效果。

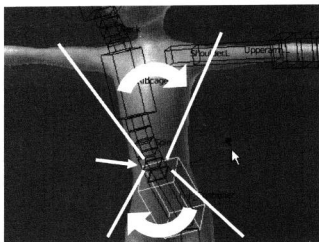


图 16-32 调整后的 Hip 骨头旋转效果

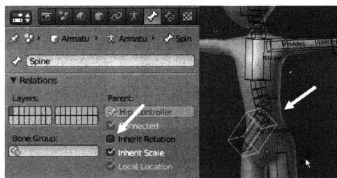


图 16-33 取消 Spine 的 Inherit Rotation (旋转关联)

#### 16.2.2.4 制作腿部约束

接下来选择大腿的 Thigh.L 骨头，如图 16-34 左图所示，然后单击 Shift 再选择 Hip 骨头，按 Ctrl+P 建立父子关系，如图 16-34 中图所示。这样在旋转 Hip 的同时，也能带动大腿 Thigh.L 骨头做旋转运动了，效果如图 16-34 右图所示。

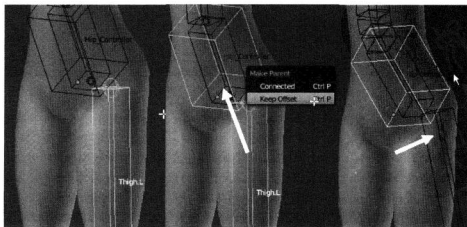


图 16-34 将大腿骨头绑定在 Hip 上

对于腿部的运动,需要为其添加一个IK约束,方便对腿部运动做快速的调节。首先选择 Shin. L 骨头,转换至骨头的约束面板中,为其添加一个IK约束修改器,将 Ankle. L 骨头设置为控制目标物体,并且修改 Chain 为 2,具体参数如图 16-35 左图所示。这样只需要控制并移动 Ankle. L 骨头,就可以带动整个腿部做弯曲运动了,效果如图 16-35 右图所示。

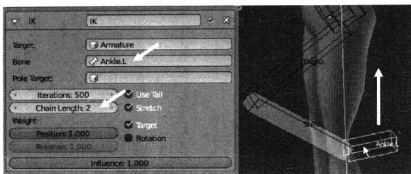


图 16-35 为腿部添加IK约束

由于IK的Chain设置为2,所以父骨头Hip的转动或移动也将带动大腿运动,如图16-36左图所示。但是由于Chain的限制效果,Hip的运动也不会将整个脚部的骨头抬起,如图16-36右图所示。

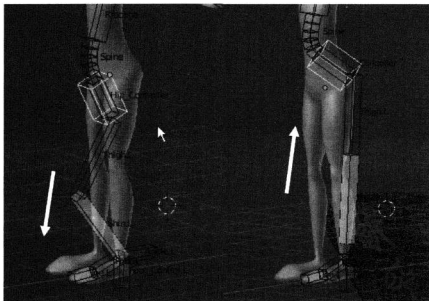


图 16-36 Hip 将受腿部IK的约束控制

接下来制作脚部的骨头绑定,首先单击RMB选择Toe(脚趾)的骨头,如图16-37左图所示。然后使用多选并单击Ctrl+P将其与Foot\_Control.L脚部控制骨头建立父子关系,如图16-37右图所示,选择保持两者之间的位置Offset(偏移量)。

拥有偏移量的父子关系骨头,会在两者之间显示出一条虚线,用于记录两者之间的关系方向,如图16-38左图箭头所示。经过这样的绑定,就可在弯曲脚掌的同时控制脚趾与地面的夹角角度了。转至Pose模式中,旋转Foot\_Control.L骨头,可以看到脚趾的骨头联动地旋转了起来,

效果如图 16-38 右图所示。

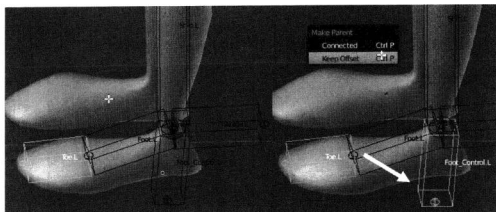


图 16-37 建立 Toe 骨头的父子关系

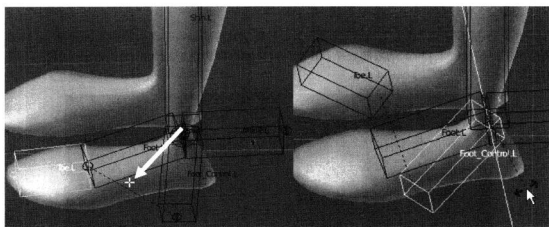


图 16-38 旋转 Foot\_Control.L 骨骼的效果

转换视图视角至脚跟位置，将骨头 Ankle.L 与 Foot.L 建立起父子关系，其中父骨头为 Foot.L 骨头，效果如图 16-39 左图所示。同时将 Foot.L 与 Foot\_Controller.L 建立父子关系，其中父骨头为 Foot\_Controller.L 骨头，效果如图 16-39 右图所示。

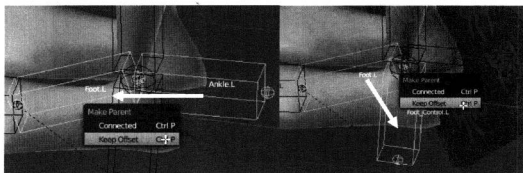


图 16-39 建立脚跟的父子关系

完成以上的绑定之后，就可以进入 Pose 模式，通过选择并调整 Foot\_Controller.L 骨头，即可

控制整个脚部的弯曲和移动等运动,如图 16-40 左图所示。如果旋转 Toe 骨头,即可在不需要固定其他物体的基础上,随意地调整脚尖与地面的关系,如图 16-40 右图所示。

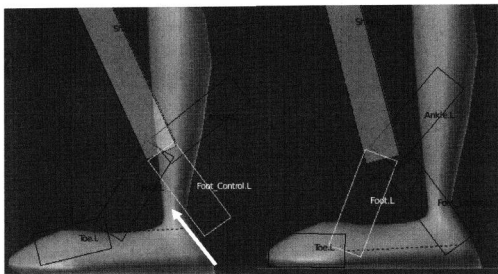


图 16-40 完整的腿部骨头运动控制

#### 16.2.2.5 制作手部约束

和制作 Hip 控制骨头的步骤类似,首先单击 Shift + D 复制出一个新的 Hand.L 骨头,如图 16-41 左图所示。使用快捷键 Ctrl + Alt + S 稍作缩放调整,如图 16-41 右图所示,将其命名为 Hand\_IK.L。

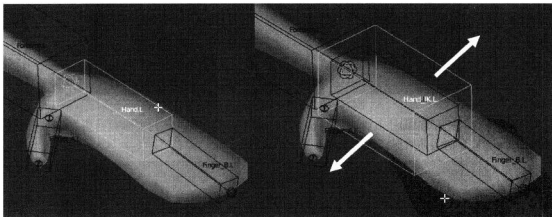


图 16-41 建立手部的控制骨头

在整体骨头的下方再添加一个新骨头,命名为 Root,用于控制整个骨架的移动,如图 16-42 左图所示。接着选择刚才新建的 Hand\_IK.L 骨头,将其与 Root 骨头建立父子关系,其中父骨头为 Root 骨头,如图 16-42 右图所示。同时也可以将 Hip 和 Foot\_Controller.L 骨头绑定为 Root 的子骨头,这样 Root 骨头就可以联动所有控制骨头做骨架的整体移动了。

由于 Hand.L 和 Hand\_IK.L 骨骼之间需要同步的运动控制,所以首先选择 Hand 骨头,在骨头约束面板中添加一个 Copy Location (坐标复制) 和一个 Copy Rotation (转动复制) 的约束修改器,如图 16-43 所示,分别用于同步控制 Hand.L 与 Hand\_IK.L 的旋转和移动运动。

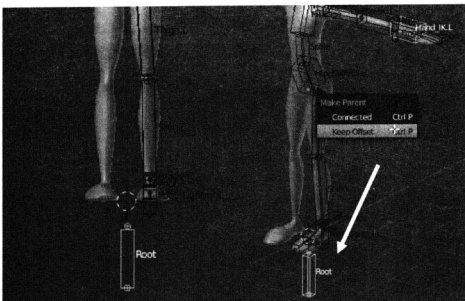


图 16-42 将 Hand\_IK.L 与 Root 建立父子关系

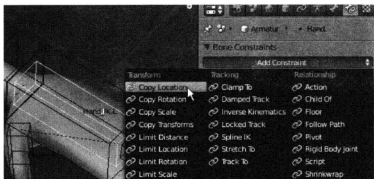


图 16-43 为 Hand.L 添加约束控制

如图 16-44 所示为 Hand.L 骨头的 Copy Location 和 Copy Rotation 约束参数面板，它们将分别读取 Hand\_IK.L 的运动数据，来实现联动约束效果。

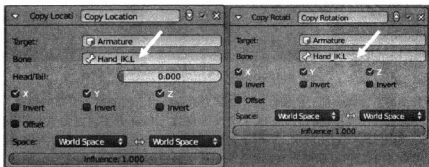


图 16-44 Copy Location 和 Copy Rotation 约束的参数面板

接着在 Pose 模式下，选择 Forearm.L 骨头，为其添加一个 IK 约束，如图 16-45 所示。

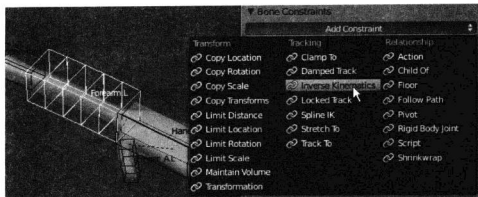


图 16-45 为 Forearm.L 骨头添加 IK 控制约束

如图 16-46 左图所示为 IK 约束的参数面板，将控制物体设置为“Hand\_IK.L”。这样就可以在选取 Hand\_IK.L 作移动和转动控制的同时，对上臂做 IK 反向运动计算。但这时会产生一个问题，那就是由于手臂的骨节属性，这里会出现一种牵连弯曲的现象，如图 16-46 右图所示。

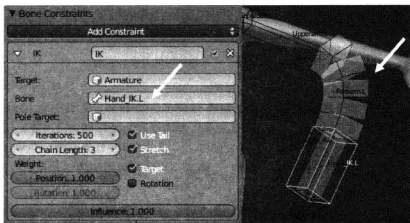


图 16-46 手臂的 IK 设置及牵连弯曲

解决的办法是调整 Forearm.L 骨头 Deform 中的骨节属性，进入其骨头属性面板，如图 16-47 左图所示，分别将 Ease In 和 Ease Out 设置为 0。这时再移动手部骨头，就可以看到，手臂骨节不会再出现弯曲现象了，效果如图 16-47 右图所示。

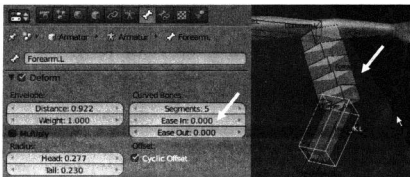


图 16-47 解决骨节的弯曲现象

但是当转动手部骨头至一个较大的弯曲角度时,如图 16-48 左图所示,手臂上的骨节又会产生一定的自旋转现象。这是因为 Upperarm.L 骨头与 Forearm.L 骨头还保持着连接状态,解决办法是在属性面板中关闭 Upperarm.L 骨头的 Connected (联接) 属性,这样无论将手臂旋转多大的角度,下臂骨头都不会出现转动现象了,如图 16-48 右图所示。

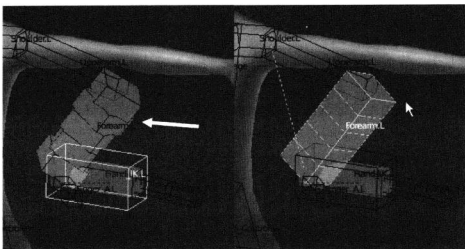


图 16-48 解决骨骼在大角度时的自旋转问题

#### 16.2.2.6 制作骨骼镜像

到此为止,左半部分的骨骼就基本上制作完成了,接下来进入骨骼的编辑模式。单击 Shift + C 把光标归位至原点处位置,然后将重心编辑模式修改为光标模式,接着使用 Shift + RMB 选择左手和左腿上的所有骨头,如图 16-49 左图所示。单击 Shift + D 复制整个左半部分骨骼,松开鼠标后,再次单击快捷键 S + X + (- Num1),将复制出的新骨骼镜像移动至角色的右半部份,如图 16-49 右图所示。

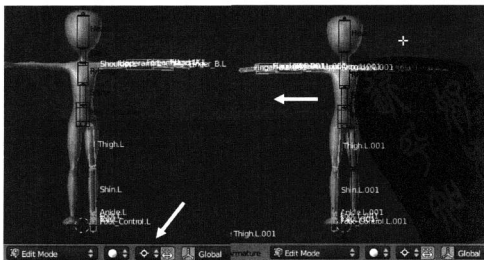


图 16-49 镜像复制骨骼



此时新复制出的骨骼会被自动添加一个后缀 .001 进行重命名, 可以在 Armature (骨骼) 面板中, 选择 Flip Names 反转命名来修改重命名的方式, 如图 16-50 左图所示, 单击后系统将取消原后缀, 并自动将名称中的“L”转换为“R”, 用于表示右侧的骨骼, 如图 16-50 右图所示。

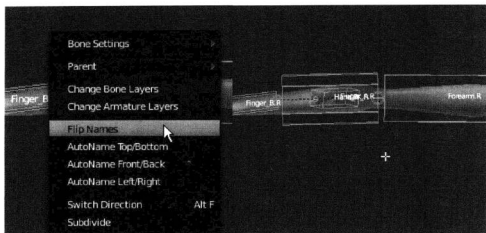


图 16-50 将骨骼做自动重命名

但是通过这种方式复制完成的骨骼镜像, 会在旋转部分骨骼时出现一些动作异常, 与左半部分的骨骼关节动作产生不匹配的对称效果。这时需要在视图窗口中单击 T 调出 Armature Option (骨骼选项), 激活其中的 X-Axis Mirror (轴镜像) 模式, 如图 16-51 所示, 这样的骨骼在变化时将不会受到镜像源的影响。

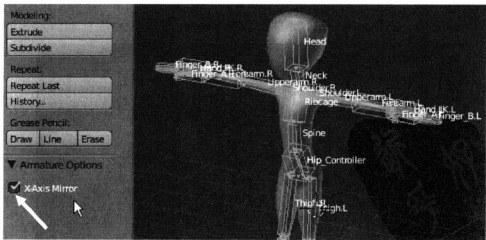


图 16-51 校正镜像物体的运动问题

完成了骨骼的约束设计, 就可以进入骨骼的 Pose (姿态) 模式为人物调整造型了。由于在骨骼的关节等位置都使用约束修改器来实现联动控制, 所以在实际操作中, 只需要对 Head、Hand\_LK、Hip、Foot\_Controller 和 Root 等几个主要控制骨头做调整和修改即可, 其余部位的骨头将依靠约束和 IK 自动完成调整计算, 如图 16-52 所示。

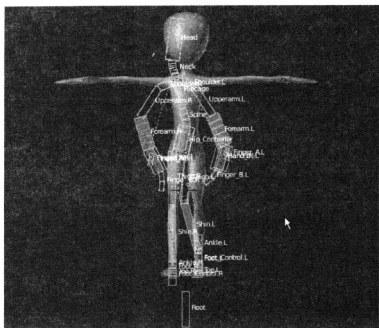


图 16-52 使用 Pose 姿势模式来调整动作

### 16.2.3 绑定模型与骨骼

接下来将制作好的骨头绑定至角色模型上，首先单击 RMB 选择角色模型，如图 16-53 左图所示。接着单击 Shift + RMB 选择骨头物体，使用快捷键 Ctrl + P 建立父子关系，在弹出的下拉菜单中选择 With Automatic Weights（使用自动权重匹配），这样模型物体将自动根据骨架上骨头的命名来新建对应部位的顶点组和权重图，如图 16-53 中图所示。单击确定完成后，角色模型就会很自然地绑定至骨骼物体上，如图 16-53 右图所示，这时就可以看到由骨骼控制的角色动作和姿态了。

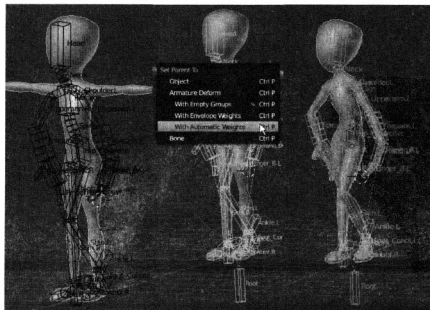


图 16-53 骨骼与模型的绑定菜单

## 16.3 制作步行动画

### 16.3.1 关键帧动画基础

#### 16.3.1.1 关键帧动画原理

动画是一种借助人眼的视觉残留原理,将静止的画面转换为动态图像的过程。在动画制作中,需要将所有的动作一一进行分解,挑选出其中的关键动作,例如抬脚离地或者是手掌敲打桌面的瞬间,并为这个动作在某个时间点上添加一个帧标识,用于设定这个动作为此刻时间点上的关键动作,而这个帧标识也就是关键帧。

帧是动画中最小单位的单幅影像画面,相当于电影胶片上的一格镜头。关键帧是一系列不连贯的动作帧分布,它们决定了整套动作的主要运动轨迹,并影响着动画的质量。关键帧之间的帧叫做过渡帧或者中间帧,它是系统运用插值的原理,自动计算出两个关键帧动作之间的运动轨迹,用来表现关键帧之间运动过渡的一种帧。在使用 Blender 等 3D 软件制作动画的过程中,只需要完成关键帧的设计,中间帧将交予系统来自动计算生成。

#### 16.3.1.2 关键帧的动作分解

接下来以一个简单的步行动画为例,来讲解如何寻找动作中的关键帧。

在制作步行动画时,可以使用一些参考图片或视频素材,去观察真实的步行动作规律,切忌凭空想象地去制作一套动作的动画,这样的效果将很难实现平滑感和真实感。这里推荐一本动画参考教材,是由 Muybridge Eadweard 撰写的《The Human Figure in Motion》,书中收录了大量的动作素材,可作为动画爱好者必备的一本标准动作参考资料。

对于步行动作,可以根据关键帧的分割原理,将一套完整的步行分为 5 个关键动作。如图 16-54 所示,为前进 1 步时 5 个动作的分解图。

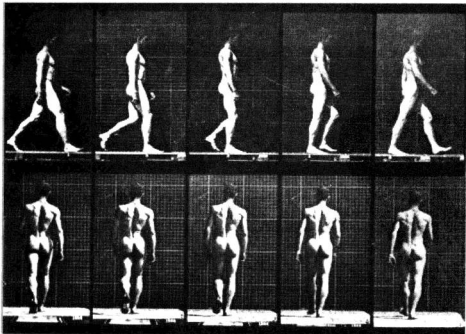


图 16-54 《The Human Figure in Motion》中的动作分解图

1) 第1步开始,人体向前提腿迈步,两腿将完全伸展,膝盖和关节处会略微弯曲,这时双脚间的距离最远。右腿在前迈的同时右臂要向后摆,左臂则应顺势前摆,接着身体的上半身会自然地向前倾。此时人体的重心在臀部,开始随着身体的前移略有下降,并朝着右脚的方向转移。

2) 接着人身体将继续向前移动,直到上身运动至右脚迈出位置的正上方,同时膝盖的关节会适当弯曲,用于吸收身体移动带来的震动和压力。两只手臂同时向身侧的两个方向回摆,身体的重心也转移至了右脚的位置,并处于步行过程中的最低点。

3) 当身体前进至当前位置的最远距离时,将自然地收回在身体后方的左腿,同时膝盖关节会伸直用于支撑起整个身体。双臂此时也完全收回至身体的两侧,并稍作停顿,身体的重心将上升至最高位置。这样就完成了一次的迈步动作,也就是步行的第一个半步。

4) 接下来的半步,人将抬起左脚,准备向前迈出,同时右脚脚跟也顺势离地,右臂停止刚才惯性的向后挥动,改为向前摆动。身体也开始下落,并随着左腿的前进稍微向前倾,重心也顺势降低。

5) 第二次向前跨步,左脚抬起向前迈,右臂前摆,重心继续降低,身体改向左脚的落地位置前移。

在制作步行动画的过程中,一定要掌握重心的变化,这是所有平衡的基础,它支持着身体的运动。同时还要注意,随着腿的前后移动,作为身体重心的臀部也将以脊椎为轴做一定的旋转,使腿部可以顺势摇摆,以保证身体的平衡。同理,肩部也要进行与臀部相反方向的转动,以保证上半身的运动平衡。而头部则需要保持在双眼注视的前进方向水平面上,通过轻微的点头和转动等动作来保持自然。

对于不同制作要求的动画,还可以对一些关键帧步骤做一定的调整。例如在制作卡通人物时,可以将第2步和第4步都省略掉,使其迈步动作看上去更加迅速而夸张。

完成一套步行流程所需要的时间,也决定了动作的快慢和回放效果。对于常规每秒24帧的影片制作来说,一些常用的完成全套步行动作的计时标准如下。

- ☐ 8 帧:非常快速的奔跑,其中每一步前进为4帧,每一秒可前进6步。
- ☐ 12 帧:非常快速的步行。
- ☐ 16 帧:卡通人物的奔跑标准。
- ☐ 24 帧:标准的步行速度,每前进一步为12帧,每一秒前进2步。
- ☐ 32 帧:懒散的步行速度。
- ☐ 40 帧:劳累或衰老时的步行速度。
- ☐ 48 帧:十分缓慢的前进。
- ☐ 64 帧:艰难的踱步前进,或者是迷路时的步行速度。

## 16.3.2 制作关键帧动画

制作关键帧动画的步骤如下。

### 16.3.2.1 准备工作

本节要制作一个人物的关键帧动画,在开始之前,可以对骨骼做一些层的重新分配。进入 Pose (姿态) 模式,首先使用 Alt + R 和 Alt + G 清空之前修改的姿态动作,然后单击 Shift + RMB 选择一些不需要人为调整的骨头,例如 Hand 和 Hip\_Control 等骨头,单击 M 弹出骨头的图层管理面板,将这些骨头移动至另外一个骨头层中,如图 16-55 所示。这样就可以减少视图中多余的骨头数量,方便对有效骨头控制器的查看与操作。

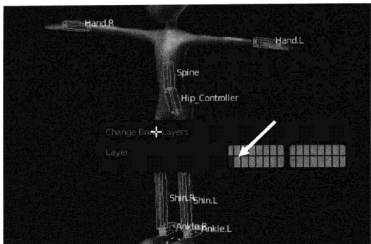


图 16-55 管理骨头层

切换至 Animation（动画）窗口布局，在 Action Editor（动作编辑器）中新建一个 Walking 动作块，如图 16-56 所示。在编辑动画时，可以将 Outline（大纲）放在侧栏，这样可以方便对多骨头做选择和调整操作。



图 16-56 Animation 动画编辑窗口


在场景中建立关键帧的方法有两种。第一种方法：可以直接在 Timeline（时间线）工具栏上开启动作记录按钮 ，这样每一次对物体执行修改，调整结果都会被 Blender 自动记录在当前的时间线上。第二种方法：也可以在完成骨骼姿态调整后，单击快捷键 I 进入关键帧菜单，如图 16-57 左图所示，选择添加 Location（位移）和 Rotation（转动）等常用的关键帧类型。在成功建立关键帧后，Action Editor（动作编辑器）中将在当前时间点上添加一个关键帧标记，如图 16-57 右图所示的黄色菱形物体。



图 16-57 添加关键帧

这两种建立关键帧方法的区别在于：前者将自动记录全部类型的的关键帧，而后者只需要记录特定类型的的关键帧。后者将产生更少的 F-Curve 曲线数据，但是这种编辑方法没有前者快速而高效。而大量冗余无用的关键帧和曲线数据会使回放速度变慢，系统将被动地消耗更多的内存资源。这里使用第一种方式来建立动画关键帧。

#### 16.3.2.2 制作步行动画帧

完成了编辑动画前的准备工作，就可以开始制作步行动画了。设定场景为 20 帧/25 帧的步行速度与回放比，这样的效果将产生一个稍快的步行效果。

首先在 Timeline（时间轴）上将当前时间标尺移动至第 1 帧的位置，然后在侧视图中调整步行动作的第一步，如图 16-58 左图所示。根据前面的分析，步行开始时，控制 Foot\_Control 骨头，让左腿为前行迈出的第一步，并且使脚跟先着地，同时调整 Toe 骨头，让脚尖轻微翘起，而右脚则应前脚掌着地，后脚跟稍微翘起。移动 Hip 骨头，使身体的重心稍微下移一点，IK 计算将自动让双腿的膝盖关节略作弯曲。手臂的摆动方向应该是向前甩动右臂，然后向后挥动左臂，肘关节稍作弯曲，手掌随着手臂的动作弧度和方向应轻微摆动，分别选择 Hand\_IK 骨头，做相应的移动和转动调整。接着转向前视图视角，可以稍微转动 Ribcage（胸部）和 Hip（胯部）的骨头，这样四肢的摆动看上去更自然，如图 16-58 右图所示。完成了基本的键动作，还可以添加其余的一些动作细节，例如转动 Head 骨头使头部微微下点，弯曲 Finger\_B 让手指向内敛等。

接着制作步行的第二个阶段，将时间轴标尺移动至第 5 帧的位置，这时身体正处于前进后，左脚落地而右脚正抬起做向前迈进的动作。如图 16-59 左图所示，控制 Foot\_Control 骨头，将左脚向后滑动至身体重心的位置，并缩小脚后跟与地面的夹角。同时将右脚抬起并移动至身体侧后方，转动 Toe 骨头使脚尖绷直。抬起 Hip 骨头，这样将适当抬高身体重心，膝盖也将自动略微伸展。移动 Hand\_IK 骨头，将双臂调节至垂直于身体两侧，同时稍微旋转 Head（头部）骨头的运动。切换至正视图视角，如图 16-59 右图所示，将 Hip 胯部骨头也向正中位置略微转动，并调节 Shoulder（肩部）和 Thigh（大腿）骨头的旋转角度。

将时间轴移动至第 10 帧，这时的动作应与第 1 帧的动作幅度相似，只是处于完全相反的镜像方向。首先将时间轴移动回第 1 帧，全选所有的骨头，单击状态栏上的复制帧工具，如图 16-60 左图箭头所示。接着重新将时间轴移动至第 10 帧，单击反向粘贴按钮，如图 16-60 右图箭

头所示，这时系统将自动把所有的帧镜像复制至当前的位置。

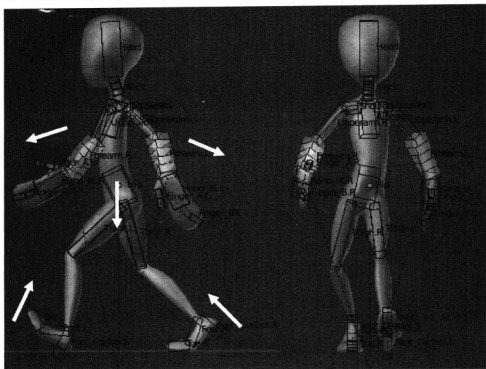


图 16-58 步行第 1 步的关键帧动作

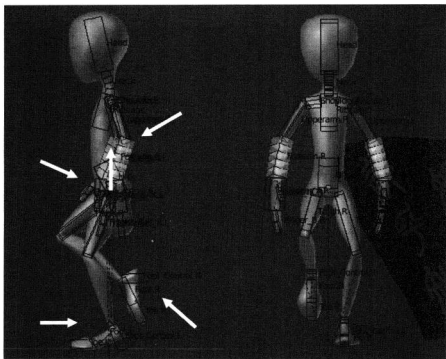


图 16-59 步行第 5 帧的关键帧动作

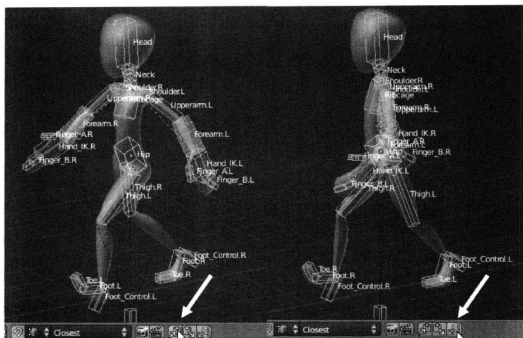


图 16-60 使用反向粘贴来制作第 10 帧的关键帧动作

同理，可以使用反向粘贴来制作第 15 帧动作。首先将时间轴定位至第 5 帧，单击帧复制按钮，如图 16-61 左图箭头所示。接着转至第 15 帧，单击反向粘贴，如图 16-61 右图箭头所示，动作即按照相反的镜像方向被粘贴至当前位置。

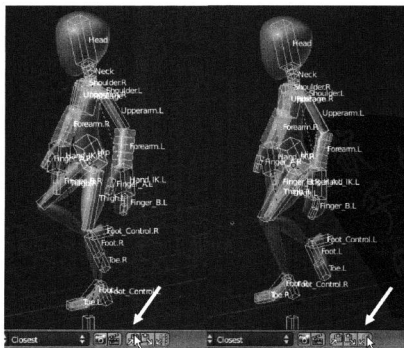


图 16-61 使用反向粘贴来制作第 15 帧的关键帧动作



接着来制作最后一个关键帧动作，这时人物完成了所有的步行动作，将回到初始状态，因此可以直接将第1帧的动作复制，使用直接粘贴的方式将帧复制到第20帧的位置，如图16-62所示。

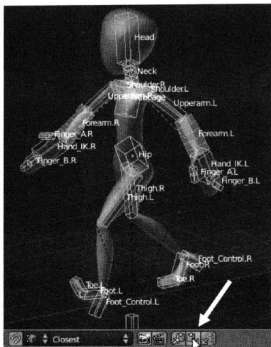


图 16-62 制作最后一帧的关键帧动作

完成了所有的关键帧设置，接着就可以单击时间轴上的回放按钮来查看动画了，或者使用快捷键 Alt + A。在播放时由于最后一帧的动作是回到第1帧的位置，所以应该将时间的播放区间设置为第1至第19帧，如图16-63所示。这样就不会在一次播放中，由于重复播放了一个关键帧动作，而造成动作出现瞬间停滞的现象。



图 16-63 设置回放区间

## 16.4 制作非线性动画

完成了单一动作段的制作，接下来就可以利用这个片段来完成连续的步行动画。首先切换窗口至 NLA（非线性动作）编辑器，当前激活物体的动作数据会自动显示为一个红色时间轴，单击轴上的小白花图标，如图16-64箭头所示，即可展开当前选择物体的动作块。

展开后的动作块将会显示在当前的 NLA 编辑区内，在 NLA 中单击 N 可调出动作段的属性面板。如图16-65所示，设置动作块的播放区间为第1~19帧，并且将 Repeat（回放）次数设置为5。这样在单击回放键 Alt + A 时，角色将自动执行5次连续的 Walking（步行）动画动作，直到第91帧的位置才停止。

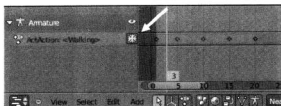


图 16-64 转至 NLA 非线性动画编辑器

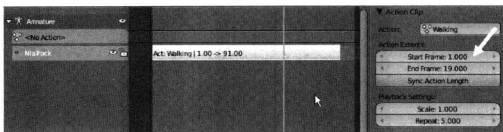


图 16-65 使用 NLA 延长并增加动作的播放时间和次数

NLA 的最大特点就是可以将多个动作做融合链接, 这里每个动作都是一个独立的动作块, 借助非线性的工作原理, 使编辑一套完整的动作流变得十分轻松。如图 16-66 所示为动画最后的回放效果, 其中曲线为身体的重心位置变化。

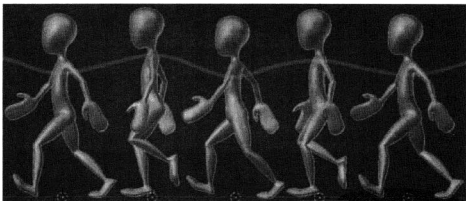


图 16-66 完成的步行动画效果

## 16.5 本章小结

本章通过一个低模角色模型制作, 简单地讲解了骨架设计、约束设计、骨头绑定、动作设计和动画制作, 接下来小结一下读者在完成本章学习后, 需要熟练掌握的基本要点。

- ❑ 动画角色模型应多使用 Loop 结构, 特别是需要大量运动的关节区域。
- ❑ 关节等处应使用相对密集的布线, 以防止弯曲变形中相邻面出现较大的扭曲现象。
- ❑ 根据实际的角色需求来设计骨头数量, 掌握使用 Segments (分段) 骨头来模拟多骨节的部位。
- ❑ 对每个骨头做适当的命名, 使用“L”或者“R”来区分左右方向。
- ❑ 掌握 IK 和父系化等功能在骨头约束设计中的应用, 熟悉基本的骨头属性调整。
- ❑ 熟悉基础的动画原理, 理解如何分解动作关节帧。
- ❑ 掌握在 Action Editor (动作编辑器) 下, 对关键帧的添加、修改、删除、镜像复制等基本操作。
- ❑ 掌握在 NLA (非线性动作) 编辑器中对动作块的常用非线性编辑操作。

## Blender大事记

版本号	日期	重要更新
v1.00	1995 年 1 月	NeoGeo 动画工作室推出开发版 Blender
v1.23	1998 年 1 月	基于 IrisGL 的 SGI 版本正式通过网络发布
v1.30	1998 年 4 月	在 Linux 和 FreeBSD 版本上成功移植 OpenGL 和 X
v1.3x	1998 年 6 月	NaN 公司成立
v1.4x	1998 年 9 月	正式发布 Sun 和 Linux Alpha 版本
v1.50	1998 年 11 月	正式发布第一本用户手册
v1.60	1999 年 4 月	带有密钥（新特性被加锁，解锁需 95 美元）的 Win 版本发布
v1.6x	1999 年 6 月	正式发布 BeOS 和 PPC 版本
v1.80	2000 年 6 月	密钥版本结束，Blender 再一次成为完全免费的软件
v2.00	2000 年 8 月	新增交互式 3D 实时引擎
v2.10	2000 年 12 月	功能更新：更新运算引擎，物理系统和 Python 支持
v2.20	2001 年 8 月	功能更新：更新人物动画系统
v2.21	2001 年 10 月	正式发布 Blender Publisher 版本
v2.2x	2001 年 12 月	正式发布 MacOSX 版本
	2002 年 10 月 13 日	Blender 正式成为开源软件，第一届 Blender 会议召开
v2.25	2002 年 10 月	正式发布 Blender Publisher 免费版
	2002 年 10 月	编码器创建了一个试验性的 Blender 代码树
v2.26	2003 年 2 月	第一次面向公众开放源码
v2.27	2003 年 5 月	第二次更新开放源码的 Blender
v2.28x	2003 年 7 月	正式发布 2.28x 系列
v2.30	2003 年 10 月	正式发布新的用户界面，并在第二届 Blender 会议上展示
v2.31	2003 年 12 月	功能更新：升级到稳定的 2.3x 用户界面系统
v2.32	2004 年 1 月	功能更新：升级内置渲染的部分功能
v2.33	2004 年 5 月	功能更新：新增 BCE 游戏引擎，AO 计算和更多内置纹理
v2.34	2004 年 8 月	功能更新：新增新的粒子系统，LSCM 的 UV 解算，Yafaray 的集成，细分表面的权重计算，渐变着色器，全屏抗锯齿等
v2.35	2004 年 11 月	功能更新：新增 Hook 物体，曲线变形工具和粒子复制
v2.36	2004 年 12 月	功能更新：强化系统稳定性，更新法向和置换贴图等大型功能
		BlenderCN 中国开源社区成立，第一版论坛上线

(续)

版本号	日期	重要更新
v2.37	2005年6月	功能更新: 修复大量的 Bug, 新增形变功能, 软体模拟, 力场模拟, 全新的细分功能, 透视阴影和多线程渲染支持
	2005年11月	BlenderCN 完成软件的中文汉化以及 Wiki 文档翻译
v2.40	2005年12月	功能更新: 修复大量的 Bug, 发布重写的动画系统, 更新形变帧功能, 毛发模拟系统, 流体模拟和骨骼系统
		BlenderCN 发布首批视频教程
v2.41	2006年1月	功能更新: 修复大量的 Bug, 更新游戏引擎的部分功能
v2.42	2006年7月	功能更新: 发布由超过 50 名开发者参与的结点系统, 新增阵列修改器, 动态模糊计算, 更新物理引擎和内置渲染器; 为 Orange 电影项目的同步开发版本
	2006年8月	电影《Elephants Dream》上映
v2.43	2007年2月	功能更新: 新增多解析细分网格, 多 UV 图层管理, 渲染通道分离模式, 烘焙工具, 雕刻工具, 拓朴工具, 新功能结点, 协同工作系统; 升级建模和动画系统, 更新喷绘笔刷, 流体系统; 重写序列编辑器, 重建官方网站; 支持多 CPU 渲染和渲染农场
v2.44	2007年5月	功能更新: 新增 SSS 着色系统, 两项修改器; 支持 64 位系统
v2.45	2007年9月	修复大量的 Bug, 改进系统性能
		BlenderCN 在清华大学承办的开源年会上展示 Blender
v2.46	2008年5月	功能更新: 发布由超过 70 名开发者参与的全新粒子与毛发系统, 更新图片浏览功能, 布料系统, AO 和烘焙功能, 动画系统的部分功能, 包括喷绘、蒙皮、约束和色彩化动作编辑器, 升级渲染中的反射效果运算性能; 新增支持肌肉系统的网格修改器, 为 Peach 电影项目的同步开发版本
v2.47	2008年8月	修复上一个版本中的 Bug
v2.48	2008年10月	功能更新: 更新 GLSL 着色器, 游戏引擎中的照明功能, Snap 吸附功能, 天空模拟, Python 编辑; 新增部分修改器, 为 Apricot 游戏项目的同步开发版本
	2008年11月	电影《Big Buck Bunny》上映
	2008年12月	游戏《Yo Frankie!》发布
		BlenderCN 发布第一期电子杂志《斑澜中国》
v2.49	2009年6月	功能更新: 更新了多项核心功能, 包括材质结点系统, 新的骨骼绘制系统, 布尔系统, 新的 Python 脚本目录, 支持 Jpeg2000 文件格式, 新的映射绘制功能; 强化了游戏引擎的部分功能, 新增视频贴图纹理, 升级了 Bullet 物理引擎, 支持鱼眼镜头渲染, 并公布了更多新的 BGE API 模块列表
	2009年7月	Ton 因为对 Blender 的杰出贡献, 被 Leeds Metropolitan University 授予了荣誉博士
v2.50	2009年9月	根据 2007 年开始的用户反馈分析, 发布 2.5x 系列的第一个版本, 正式进入全新内核和用户界面的测试阶段; 新增多窗口多场景显示模式, RNA 数据结构, 实体化等修改器和新的约束, 支持自定义快捷键, 体积着色, 色彩管理系统; 更新动画模块, 雕刻功能, 绘制工具, 烟雾等物理模拟系统, 粒子系统, 光线追踪系统; 升级对 64 位系统的支持和部分工具的选项参数
v2.52	2010年3月	更新并完善全部用户界面, 修复了超过 200 个 Bug
		BlenderCN 成都/北京工作室正式成立, 论坛改版, 开通译园博客, 考评系统正式上线
v2.53	2010年5月	功能更新: 修改了 API 库; 更新了控制台系统, 图片编辑系统, 网络渲染平台; 新增间接光照模式; 修复了 347 个 Bug
v2.54	2010年8月	发布的一个较稳定的 2.5 系列版本, 修复了 360 个 Bug
	2010年10月	Blender 第三部电影《Sintel》上映
v2.56	2010年12月	强化了系统稳定性, 修复了 440 个 Bug, 开始启动 Wiki 文档更新项目, 为 Durian 电影项目的同步开发版本
v2.6x	2011年	.....

## Blender默认快捷键

模 式	快 捷 键	说 明
通 用	RMB	选择操作
	Shift + RMB	多选操作
	MMB 拖动	转动视图操作
	MMB 滚动	缩放操作
	Shift + MMB 拖动	平移视图操作
	Ctrl + MMB 拖动	缩放视图操作
	LMB	定位光标
	Shift + A	新建物体
	Ctrl + LMB	快速新建物体
	X	删除操作
	SpaceBar	搜索操作
	T	弹出工具菜单
工 程	N	弹出属性菜单
	Ctrl + N	新建工程
	Ctrl + O/F1	打开工程
	Shift + Ctrl + O	打开最近的工程
	Ctrl + Q	退出
	Ctrl + S/W	保存
	Shift + Ctrl + S/F2	另存为
	F3	保存渲染结果
	F8	重载脚本
	F11	显示渲染结果
	F12	渲染当前静帧
	Ctrl + F12	渲染动画
	W	仅渲染选择的物体
	Shift + B	仅渲染框选部分
	ESC	取消渲染
	Ctrl + Z	撤销操作
	Ctrl + Shift + Z/F6	重做操作
	Ctrl + Alt + U	打开配置面板
	Ctrl + U	保存修改为默认设置
	J	循环预览渲染缓存
	U	物体独立化
	Ctrl + Alt + P	物体代理模式

(续)

模 式	快 捷 键	说 明
工 程	Shift + F1	导入外部库中的物体
	Ctrl + F3	快速截屏
	Alt + F3	录制截屏
视图/导航	Ctrl + Alt + Q	切换至全四视图模式
	Ctrl + UP	将当前窗口最大化
	Alt + F11	全屏模式
	Num1	前视图
	Num3	右视图
	Num7	顶视图
	Ctrl + Num1	后视图
	Ctrl + Num3	左视图
	Ctrl + Num7	底视图
	Num5	切换透视/正交视图
	Num0	相机视图
	Shift + B	放大视图至框选区域
	Ctrl + 0	激活当前镜头为场景的默认镜头
	Ctrl + Alt + Num0	将相机切换至当前视图
	Home	重置视图
图 层	.	放大视图至当前选择的物体
	+ / -	步进放大/缩小视图
	~	显示全部图层
镜头飞行模式	1/2/3/4/5/6/7/8/9/0	快捷切换前 10 个图层
	Shift + F	开启镜头飞行模式
	MMB 前滚	加速
	MMB 回滚	减速
	MMB 单击	悬停
	W	向前飞行
	S	向后飞行
	A	左转飞行
	D	右转飞行
	R	爬升
模式转换	F	下降
	TAB	物体/编辑模式切换
	V	物体/顶点绘制模式切换
	Ctrl + TAB	物体/权重绘制模式切换
	Ctrl + Left/Right Arrow	窗口布局切换
	Shift + F2	切换至逻辑编辑器
	Shift + F3	切换至结点编辑器
	Shift + F4	切换至控制台
	Shift + F5	切换至 3D 视图
	Shift + F6	切换至曲线编辑器
	Shift + F7	切换至工具面板
	Shift + F8	切换至视频序列编辑器
	Shift + F9	切换至大纲编辑器
	Shift + F10	切换至 UV 图片编辑器
	Shift + F11	切换至文本编辑器

(续)

模 式	快 捷 键	说 明
编辑模式：选择	A	全选/取消全选操作
	B	框选操作
	C	圈选操作
	Ctrl + LMB	鼠标悬停圈选
	Alt + RMB	点/边环选操作
	Ctrl + Alt + RMB	面环选操作
	Alt + RMB	选择叠加的物体
	L	关联选择操作
	Shift + G	选择群组物体
	Shift + Ctrl + M	选择镜像物体
	Ctrl + L	全部关联选择操作
	Ctrl + I	反选操作
	O	切换衰减编辑模式
编辑模式：形变	Shift + Tab	切换吸附编辑模式
	X	调用 X 轴
	Y	调用 Y 轴
	Z	调用 Z 轴
	G	移动操作
	R	转动操作
	S	缩放操作
	Alt + G	清除移动形变
	Alt + R	清除旋转形变
	Alt + S	清除缩放形变
	G/R/S + Shift	低步进值操作
	G/R/S + Ctrl	按照 Blend 单位步进操作
	MMB + X/Y/Z	锁定对应坐标轴
	Shift + W	按光标位置缠绕
	Shift + Alt + S	球面化
	Shift + Ctrl + Alt + S	交错形变
编辑模式：网格工具	Ctrl + A	应用形变
	E	挤压操作
	E + RMB	在当前位置挤压
	Shift + E	仅挤压出边
	Shift + D	复制
	Alt + D	关联复制
	Ctrl + J	合并物体
	P	分离为新物体
	Ctrl + H	新建钩键物体
	W	细分操作
	Ctrl + E	标记缝合边
	Alt + M	点合并操作
	Ctrl + M	镜像操作
	Alt + S	膨胀缩放操作
	F	缝合面操作
	Alt + F	强制缝合操作
	Shift + Alt + F	完美缝合操作

(续)

模 式	快 捷 键	说 明
编辑模式：网格工具	Ctrl + R	环切操作
	K + LMB	精确切割
	Shift + K + LMB	精确二分切割
	V	分割工具
	Ctrl + 1/2/3/4	快速细分操作
	Ctrl + N	计算网格表面的法线
	Shift + Ctrl + N	反向计算网格表面的法线
	Alt + C	物体类型转换
	Z	切换实体与线框显示模式
编辑模式：特殊菜单	Alt + Z	切换实体与光影显示模式
	Ctrl + V	弹出点菜单
	Ctrl + E	弹出边菜单
	Ctrl + F	弹出面菜单
	Ctrl + H	弹出钩键菜单
	Shift + S	弹出吸附菜单
	Ctrl + TAB	网格编辑模式选择菜单
	U	弹出 UV 解算菜单
	W	弹出特殊选项菜单
编辑模式：关系	Shift + Ctrl + Tab	弹出吸附模式菜单
	M	移动物体至指定层
	Ctrl + M	移动物体至镜像坐标位置
	H	隐藏选择物体
	Alt + H	取消隐藏
	Ctrl + Shift + Alt + C	原点操作选项菜单
	Ctrl + P	建立父子关系操作
	Alt + P	取消父子关系操作
	Ctrl + T	建立跟踪关系操作
	Alt + T	取消跟踪关系操作
	Shift + C	重置光标位置至原点位置
	Ctrl + G	群组操作
	Ctrl + Alt + G	将当前物体从群组中移除
	Shift + Ctrl + G	将当前物体添加至激活群组中
	Shift + Alt + G	将当前物体从激活群组中移除
	Shift + Ctrl + C	添加约束关系
	Ctrl + Alt + C	清除约束关系
编辑模式：曲线	H	隐藏手柄显示
	Alt + H	取消隐藏手柄显示
	V	切换矢量手柄控制模式
	Shift + H	隐藏未选手柄显示
	Alt + C	封闭路径操作
	Ctrl + LMB	添加控制节点
	W	细分操作
	Ctrl + T	节点倾角调整操作
	Alt + T	清除倾角操作
雕刻/绘制模式	F	调节笔刷尺寸
	Shift + F	调节笔触强度
	Ctrl + F	旋转笔刷纹理



(续)

模 式	快 捷 键	说 明
雕刻/绘制模式	Shift + K	调节笔刷权重
画线工具	D + LMB	自由绘制
	Ctrl + D + LMB	直线绘制
	D + RMB	擦除工具
UV 编辑	V	缝合工具
	P	图钉工具
	Alt + P	取消当前图钉
	E	解算 UV
	Ctrl + V	自适应缩放调节
	Ctrl + P	最优化显示排列
	Ctrl + A	自动缩放
动 画	Alt + A	回放动画
	Alt + Shift + A	反向回放动画
	P	设置预览帧区间
	Alt + P	取消预览帧区间
	Right Arrow	跳跃至下一帧
	Left Arrow	跳跃至前一帧
	Up Arrow	向前跳跃 10 帧
	Down Arrow	向后跳跃 11 帧
	Shift + Left Arrow	跳跃至第一帧
	Shift + Right Arrow	跳跃至最后一帧
	Alt + MMB 滚动	单帧浏览
	I	新建关键帧
	Alt + I	移除关键帧
	Ctrl + Page Up	跳跃至下一个关键帧
	Ctrl + Page Down	跳跃至上一个关键帧
编辑模式: 骨骼	E/Ctrl + LMB	新建骨骼操作
	Ctrl + R	自旋骨骼操作
	Ctrl + N	重新计算骨骼法线操作
	Ctrl + Alt + A	重新排列骨骼操作
	Ctrl + Alt + P	分离骨骼
	M	将骨骼移动至指定的层
	Shift + W	弹出骨骼标识菜单
	Alt + F	反向骨骼操作
	] / [	选择继承关系的上/下级的骨骼
	Shift + ] / [	多选继承关系的上/下的骨骼
	L	选择连接的骨骼
姿势模式	Ctrl + A	应用骨骼的形变操作结果
	Alt + R	清除骨骼的旋转形变
	Alt + G	清除骨骼的位移形变
	Alt + S	清除骨骼的缩放形变
	Ctrl + C	复制当前动作
	Ctrl + V	将缓存中的动作粘贴至当前帧
	Shift + I	添加 IK 约束
	Ctrl + Alt + I	移除 IK 约束
	Ctrl + G	群组骨骼操作

(续)

模 式	快 捷 键	说 明
姿势模式	Alt + E	放松骨骼动作
时间线	S	设置初始帧的位置
	E	设置结束帧的位置
	Home	显示全部帧
	M	添加标记符
	G	移动标记符
	Ctrl + T	帧/秒显示模式切换
视频编辑	Page Up	跳跃至下一个素材段
	Page Down	跳跃至上一个素材段
	K	按照时间帧对素材段做切割操作
	Shift + L	锁定当前素材段
	Shift + Alt + L	解锁当前素材段
	Ctrl + C	复制素材段
	Ctrl + V	将缓存中的素材段粘贴至当前帧
结点编辑器	Shift + S	将素材段吸附至当前时间线位置
	Shift + A	添加一个新结点
	Ctrl + LMB	分离连接结点
	H	隐藏当前选择的结点
	M	静默当前选择的结点
	Ctrl + G	群组当前选择的结点
	Alt + G	取消群组属性
	TAB	群组结点的编辑模式
	Alt + MMB 拖动	移动背景图操作
	V	放大背景图操作
	Alt + V	缩小背景图操作